

BIBLIOTECA PRACTICA

TALLER DE INFORMATICA

TRUCOS - SPRITES
BRICOLAGE DEL HARD
AULA ABIERTA
LOGO - TERMINOLOGIA

PROGRAMAS VALIDOS PARA AMSTRAD,
IBM, SPECTRUM, COMMODORE Y MSX



PROGRAMAS PRACTICOS

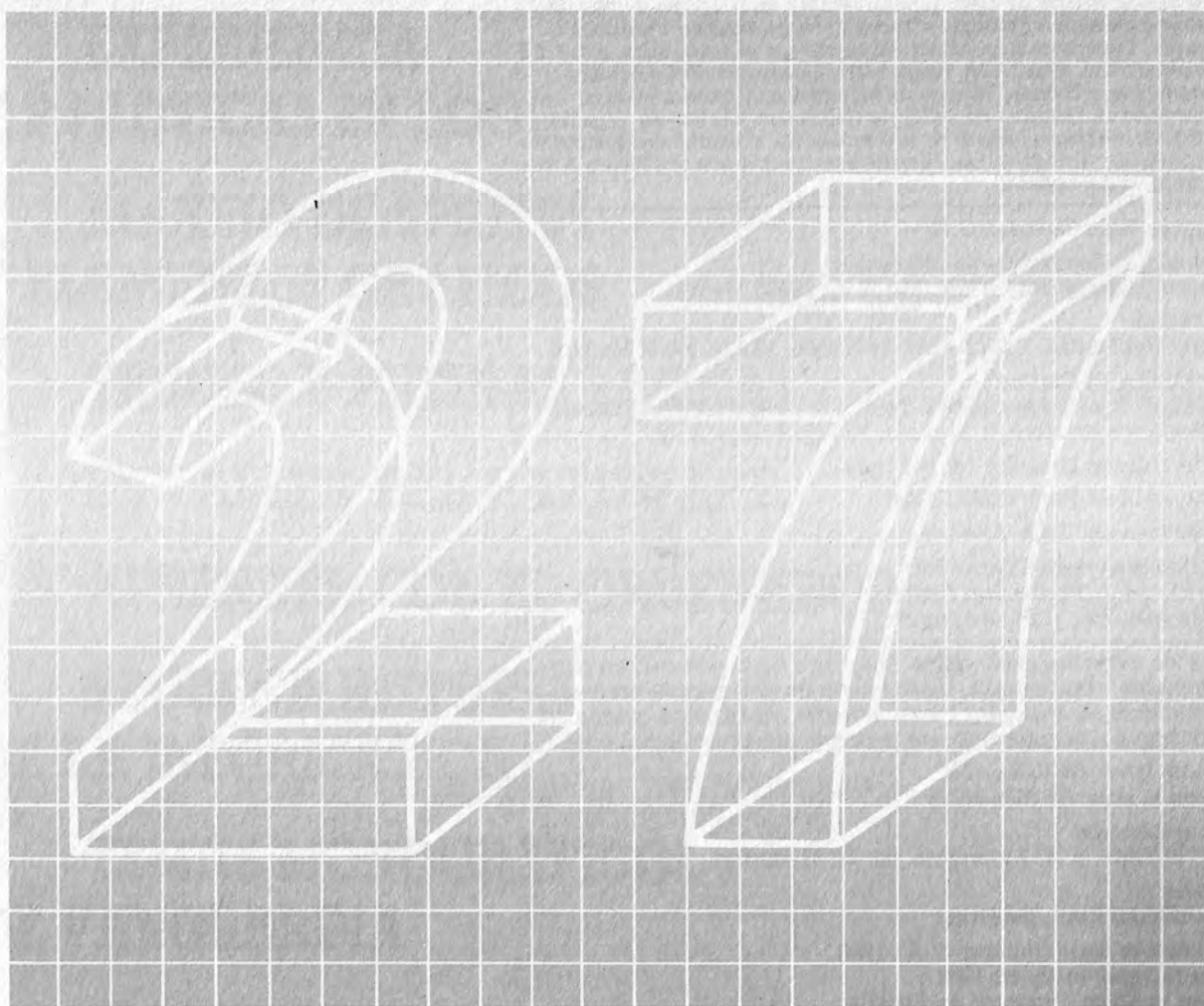
TRUCOS, BRICOLAGE Y GRAFICOS

EDICIONES SIGLO CULTURAL

BIBLIOTECA PRACTICA TALLER DE INFORMATICA

TRUCOS - SPRITES
BRICOLAGE DEL HARD
AULA ABIERTA
LOGO - TERMINOLOGIA

PROGRAMAS VALIDOS PARA AMSTRAD,
IBM, SPECTRUM, COMMODORE Y MSX



EDICIONES SIGLO CULTURAL

Una publicación de

EDICIONES SIGLO CULTURAL, S. A.

Director-editor:

RICARDO ESPAÑOL CRESPO.

Gerente:

ANTONIO G. CUERPO.

Directora de producción:

MARIA LUISA SUAREZ PEREZ.

Director de la colección:

JOSE ARTECHE, Ingeniero de Telecomunicación

Diseño:

BRAVO-LOFISH.

Maquetación:

D. S. M.

Dibujos:

JOSE OCHOA

Tomo 27. «Experiencias y prácticas en logo», Carlos Albert, Técnico de Informática; Adoración Llena Jubero, Técnico de Informática. «Manejo de sprites y elementos gráficos», «Trucos y rutinas básicas», Francisco Morales. Técnico de Informática. «Aprende con el ordenador», AULA DE INFORMATICA APLICADA: Angel Leira, Licenciado de Geografía e Historia; Jorge Thomas, Técnico de Informática; Alejandro Marcos, Licenciado en Química, «El Taller de Hardware», Carlos Rey, Ingeniero Industrial. «Pequeña historia de la Informática», «Temas monográficos de vanguardia», «Terminología», «Vocabulario de Informática», Blanca Arbizu, Técnico de Informática.

Ediciones Siglo Cultural, S. A.

Dirección, redacción y administración:

Pedro Teixeira, 8-2.º. Teléf. 810 52 13. 28020 Madrid.

Publicidad:

Gofar Publicidad, S. A. Benito de Castro, 12 bis. 28028 Madrid.

Distribución en España:

COEDIS, S. A. Valencia, 245. Teléf. 215 70 97. 08007 Barcelona.

Delegación en Madrid: Serrano, 165. Teléf. 411 11 48.

Distribución en Ecuador: Muñoz Hnos.

Distribución en Perú: DISELPESA.

Distribución en Chile: Alfa Ltda.

Importador exclusivo Cono Sur:

CADE, S.R.L. Pasaje Sud América, 1532. Teléf.: 21 24 64.

Buenos Aires - 1.290. Argentina.

Todos los derechos reservados. Este libro no puede ser, en parte o totalmente, reproducido, memorizado en sistemas de archivo, o transmitido en cualquier forma o medio, electrónico, mecánico, fotocopia o cualquier otro, sin la previa autorización del editor.

ISBN del tomo: 84-7688-147-9

ISBN de la obra: 84-7688-047-2

Fotocomposición:

ARTECOMP, S. A. Albarracín, 50. 28037 Madrid.

Imprime:

GRAFICAS EMA. (Madrid)

© Ediciones Siglo Cultural, S. A., 1986

Depósito legal: M-43.185-1986

Printed in Spain - Impreso en España.

Suscripciones y números atrasados:

Ediciones Siglo Cultural, S. A.

Pedro Teixeira, 8-2.º. Teléf. 810 52 13. 28020 Madrid.

Julio, 1987

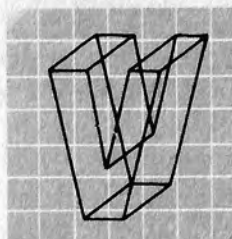
P.V.P. Canarias: 355.—Ptas.

INDICE

■ EXPERIENCIA Y PRACTICAS EN LOGO	5
■ MANEJO DE SPRITES Y ELEMENTOS GRAFICOS	15
■ TRUCOS Y RUTINAS BASICAS	24
■ EL TALLER DEL HARDWARE	32
■ APRENDER CON EL ORDENADOR	38
■ PEQUEÑA HISTORIA DE LA INFORMATICA	50
■ TEMAS MONOGRAFICOS DE VANGUARDIA	54
■ TERMINOLOGIA	58
■ VOCABULARIO DE INFORMATICA	59

EXPERIENCIA Y PRACTICAS EN LOGO

Operaciones con objetos



ACIO? objeto.

Devuelve CIERTO si el objeto es la palabra vacía o la lista vacía, si no, devuelve FALSO.

? ESCRIBE VACIO? "
CIERTO

? ESCRIBE VACIO? "A
FALSO

? ESCRIBE VACIO? []
CIERTO

? ESCRIBE VACIO? [ALGO]
FALSO

NOTA: En el LOGO del SPECTRUM la sintaxis de esta primitiva es:

ESTA.VACIA? objeto

— PRIMERO objeto

abreviada:

PRI objeto.

Esta primitiva devuelve el primer elemento del objeto especificado.

Si el objeto es una palabra devuelve la primera letra.

? ESCRIBE PRIMERO "CASA
C

? ESCRIBE PRIMERO "A
A

PRIMERO no acepta como dato la palabra vacía.

? ESCRIBE PRIMERO "
PRI NO ACEPTA COMO DATO

Si el objeto es un número, debemos tener en cuenta los siguientes casos:

? ESCRIBE PRI ".5

? ESCRIBE PRI .5
0

Si el objeto es una lista, devuelve la primera palabra o la primera lista.

? ESCRIBE PRIMERO [ESTO ES MUY SENCILLO] ESTO

? ESCRIBE PRIMERO [HOLA]
HOLA

? ESCRIBE PRIMERO [[ENERO FEBRERO MARZO] TRIMESTRE]
ENERO FEBRERO MARZO

El objeto especificado no puede ser la lista vacía.

? ESCRIBE PRIMERO []
PRI NO ACEPTA [] COMO DATO

— ULTIMO objeto

abreviada:

UL objeto

Devuelve el último elemento del objetivo especificado.

Si el objeto es una palabra devuelve la última letra.

? ESCRIBE ULTIMO "CASA
A

? ESCRIBE ULTIMO "A
A

Logo reconoce cuándo dos objetos son iguales o no.

EXPERIENCIA Y PRACTICAS EN LOGO

? ESCRIBE UL "
UL NO ACEPTA COMO DATO

Particularidades si el objeto es un número:

? ESCRIBE UL "1.20
0
? ESCRIBE UL 1.20
2

Si el objeto es una lista, devuelve la última palabra o la última lista.

? ESCRIBE UL [RAPIDO Y SENCILLO]
SENCILLO
? ESCRIBE UL [MAS RAPIDO [MAS SENCILLO] [SENCILLISIMO]]
SENCILLISIMO

? ESCRIBE UL []
UL NO ACEPTA [] COMO DATO

— IGUALES objeto1 objeto2

Devuelve CIERTO si objeto1 y objeto2 son iguales, si no, devuelve FALSO.

? ESCRIBE IGUALES? "A "A
CIERTO

? ESCRIBE IGUALES? ".1 "0.1
CIERTO

? ESCRIBE IGUALES? [DOS PALABRAS]
[DOS PALABRAS]
CIERTO

Podemos utilizar la escritura infija de esta operación.

? ESCRIBE "AS = "AS
CIERTO

El siguiente procedimiento escribe una palabra que empieza por la letra que introducimos:

(Sólo letras de la A a la E)

? PARA LETRAS
> HAZ "C LC
> SI :C = "A [ESCRIBE "AMERICA]
> SI :C = "B [ESCRIBE "BARCO]
> SI :C = "C [ESCRIBE "CASA]
> SI :C = "D [ESCRIBE "DADO]
> SI :C = "E [ESCRIBE "ESCUELA]
> FIN

Si ejecutamos el procedimiento LETRAS y pulsamos la tecla A, en la pantalla aparecerá la palabra AMERICA.

? LETRAS
AMERICA

NOTA: En el LOGO del SPECTRUM la sintaxis de esta primitiva es:

ES.IGUAL? objeto1 objeto2

■ Gestión de archivos (sólo PC-COMPATIBLES)

A continuación explicamos la función de cuatro primitivas que podemos encontrar en algunas versiones de LOGO para los PC-COMPATIBLES.

— PONPOSESCRITURA *n*

Coloca el puntero en la posición *n* del archivo que en ese momento esté abierto para la escritura.

En número *n* indica la posición del puntero en el archivo, en octetos, puede variar entre 0 y la longitud total del archivo.

? ABRE "TEXTO
? PONESCRITURA "TEXTO
? PONPOSESCRITURA 10
? ESCRIBE [PRUEBA 1]
? PONPOSESCRITURA 20
? ESCRIBE [PRUEBA 2]
? CIERRA "TEXTO

Abrimos el archivo TEXTO para escritura, colocamos el puntero en el décimo octeto y escribimos PRUEBA 1, colocamos el puntero en el octeto 20 y escribimos PRUEBA 2.

Recuerda que un octeto equivale a un carácter.

Comprobemos el contenido del archivo TEXTO desde el sistema operativo.

A> TYPE TEXTO
PRUEBA 1
PRUEBA 2

— PONPOSLECTURA *n*

Coloca el puntero en la posición *n* del fichero abierto para lectura en ese momento.

El número *n* (número de octeto) puede variar entre 0 y la longitud total del archivo.

La primitiva PRIMERO devuelve el primer elemento del objeto especificado.

```
? ABRE "TEXT0
? PONLECTURA "TEXT0
? PONPOSLECTURA 0
? HAZ "C LL
? HAZ "B LL
? CIERRA "TEXT0
? ESCRIBE :C
? ESCRIBE :B
```

En la pantalla aparece:

```
PRUEBA 1
PRUEBA 2
```

Las dos listas que anteriormente hemos guardado en el archivo TEXT0.

Veamos otro ejemplo utilizando PONPOSESCRITURA y PONPOSLECTURA.

```
? PARA P1
> ABRE "TEXT0
> PONESCRIPTURA "TEXT0
> PONPOSESCRITURA 8
> ESCRIBE "LOLA
> PONPOSESCRITURA 14
> ESCRIBE "PPP
> CIERRA "TEXT0
> FIN
```

```
? PARA P2
> ABRE "TEXT0
> PONLECTURA "TEXT0
> ESCRIBE LL
> ESCRIBE LL
> CIERRA "TEXT0
> FIN
```

Ejecutamos primero el procedimiento que escribe datos en el archivo TEXT0.

```
? P1
```

Los datos ya se habrán guardado en el archivo, leámoslos con el procedimiento P2.

```
? P2
```

En la pantalla aparece:

```
PRUEBA ILOLA
PPP
```

```
— POSESCRITURA
```

Esta primitiva devuelve la posición del puntero en el archivo actualmente abierto para escritura.

```
? ABRE "TEXT01
? PONESCRIPTURA "TEXT01
? PONPOSESCRITURA 0
? HAS "A PONESCRIPTURA
? ESCRIBE "HOLA
? HAZ "B POSESCRITURA
? CIERRA "TEXT01
? ESCRIBE :A ESCRIBE :B
```

El contenido de la variable A es cero y el de la variable B seis.

Hemos asignado la posición del puntero a dos variables que más tarde hemos consultado, porque si hubiésemos utilizado la orden ESCRIBE, los dos números se habrían escrito en el archivo TEXT01 y no en la pantalla como queríamos.

```
— POSLECTURA
```

Devuelve la posición del puntero en el archivo actualmente abierto para lectura.

```
? ABRE "TEXT01
? PONLECTURA "TEXT01
? ESCRIBE POSLECTURA
```

La posición del puntero es 0.

■ Procedimientos en grupos

Algunas versiones de Logo dispone de una serie de primitivas para manejar los procedimientos en grupos.

La primitiva que a continuación vamos a exponer sólo podemos encontrarlas en algunas versiones del Logo para los PC-COMPATIBLES.

```
— AGRUPAR PROCEDIMIENTOS
```

Existe una orden con la que podemos crearnos diferentes grupos de procedimientos:

```
? AGRUPA "grupo [lista procedimientos]
```

Esta orden agrupa bajo el nombre del grupo que se especifica, los procedimientos que se señalan en la lista.

De esta forma podemos crearnos subconjuntos en el área de trabajo de la memoria del ordenador y tener de forma más estructurada y organizada los procedimientos definidos.

Si el objeto que acompaña a la primitiva ULTIMO es una palabra, la primitiva devuelve la última letra de la palabra.

EXPERIENCIA Y PRACTICAS EN LOGO

El tener los procedimientos agrupados nos permite también guardar en archivos un grupo determinado de procedimientos, sin necesidad de especificar el nombre de cada uno de ellos, sino especificando exclusivamente el nombre del grupo.

Supongamos que tenemos definidos en memoria los siguientes procedimientos:

RAIZ
TRONCO
RAMA
HOJA
PERRO
GATO
PAJARO
PATO

Dependiendo de la relación que guardan los procedimientos, podríamos definir dos grupos, uno que podríamos llamar ARBOL y otro ANIMALES:

? AGRUPA "ARBOL [RAIZ TRONCO RAMA HOJA]

? AGRUPA "DOMESTICOS [PERRO GATO PAJARO]

Ahora podemos guardar por separado los grupos creados:

? GUARDA "PLANTAS "ARBOL

Los procedimientos del grupo ARBOL se guardan en el archivo PLANTAS.

? GUARDA "ANIMALES "DOMESTICOS

Los procedimientos del grupo DOMESTICOS se guardan en el archivo ANIMALES.

Otra primitiva que nos permite agrupar procedimientos es:

.. ? AGTODO "grupo

Abreviatura de AGrupa TODO.

Se agruparán bajo el nombre de grupo que se especifica todos los procedimientos y variables (con sus contenidos), que se encuentren en el área de trabajo de la memoria.

Si alguno de los procedimientos está ya en un grupo, no podrá formar parte de éste.

— BORRAR GRUPOS

El Logo nos permite borrar los grupos que tengamos creados. Esto lo conseguimos gracias a la orden:

? BOG "grupo

Abreviatura de BORra Grupo.

Borra el grupo que se especifica del área de trabajo de la memoria. Es decir, todos los procedimientos que componen el grupo que se especifica se borrarán.

— IMPRIMIR GRUPOS

Los grupos que tenemos formados podemos imprimirlos bien sea en la pantalla o en la pantalla y en la impresora.

La orden que nos permite esto es:

? IM "grupo

Los procedimientos que componen el grupo que se especifica se presentarán en la pantalla del ordenador.

Si la impresora está conectada y activado el canal de comunicación, obtendremos también en papel el listado de los procedimientos.

— EDITAR GRUPOS

Al igual que podemos presentar en el editor uno o varios procedimientos, también podemos ordenar que sea un grupo el que se edite.

Esto lo conseguimos con la orden:

? EDG "grupo'

Abreviatura de EDita Grupo.

Los procedimientos que componen el grupo que se especifica se presentará en el editor Logo.

■ Cuadro resumen

— VACIO? objeto

Devuelve CIERTO si objeto es la palabra o la lista vacía, si no, devuelve FALSO.

En el Logo del SPECTRUM la sintaxis es: ESTA.VACIA? objeto

— PRIMERO objeto

Devuelve el primer elemento de objeto.

Si el objeto es una palabra, devuelve la primera letra, si es una lista, devuelve la primera palabra o la primera lista.

— ULTIMO objeto

Devuelve el último elemento de objeto.

Las primitivas PRIMERO y ULTIMO no admiten la palabra vacía ni la lista vacía.

Si objeto es una palabra, devuelve la última letra, si es una lista, devuelve la última palabra o la última lista.

— IGUALES? objeto1 objeto2

Devuelve CIERTO si objeto1 y objeto2 son iguales, si no, devuelve FALSO.

En el LOGO del SPECTRUM la sintaxis es:

ES.IGUAL? objeto1 objeto2

(Sólo PC-COMPATIBLES)

— PONPONESCRITURA *n*

Coloca el puntero en la posición *n* del archivo actualmente abierto para escritura.

— PONPOSLECTURA *n*

Coloca el puntero en la posición *n* del archivo actualmente abierto para lectura.

— POSESCRITURA

Devuelve la posición del puntero en el archivo actualmente abierto para escritura.

— POSLECTURA

Devuelve la posición del puntero en el archivo actualmente abierto para lectura.

— ? AGRUPA "grupo [lista procedimiento]

Los procedimientos que se especifican en la lista se agruparán bajo el nombre de grupo que se indica.

— ? AGTODO "grupo

Se agruparán bajo el nombre de grupo que se especifica todos los procedimientos y variables que haya en el área de trabajo de la memoria.

— ? BOG "grupo

Se borrará del área de trabajo los procedimientos que componen el grupo que se especifica.

— ? IM "grupo

Los procedimientos que componen el grupo que se especifica se presentarán en la pantalla del ordenador y en la impresora (de estar ésta activada).

— ? EDG "grupo

Entra en el editor Logo y muestra todos los procedimientos que componen el grupo que se especifica.

Ejercicios

1. Define los procedimientos necesarios para realizar, hasta que se introduzca la palabra *FIN*, lo siguiente:
 - a) Leer una palabra del teclado.
 - b) Escribir en la pantalla la primera y la última letra de la palabra tecleada.
2. Haz lo mismo que en el ejercicio anterior, pero en vez de con palabras con listas.
3. Realiza el siguiente dibujo.

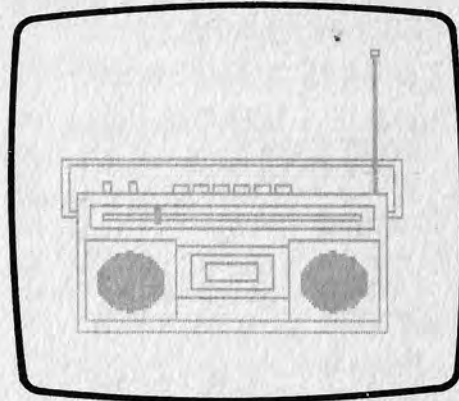


Fig. 1.

4. Define los procedimientos necesarios para dibujar en la pantalla lo que muestra la figura siguiente:

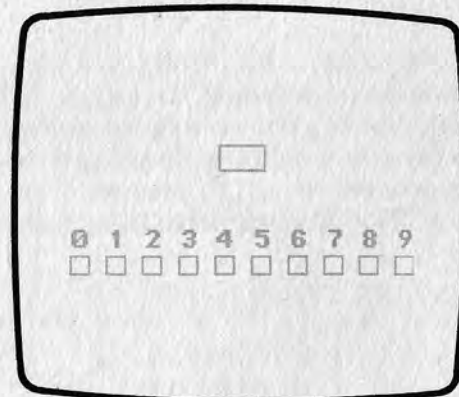


Fig. 2.

Una vez hecho esto, tienes que para señalar con una flecha el número al azar asignado a una variable, a la vez que el número aparece el marcador superior. Este proceso tiene que realizarse 100 veces.

Algunas versiones del Logo permiten crear grupos de procedimientos.

EXPERIENCIA Y PRACTICAS EN LOGO

5. ¿Son correctas las órdenes siguientes?

- ? ABRE "TABLA
- ? PONLECTURA "TABLA
- ? CIERRA "TABLA
- ? ESCRIBE POSLECTURA
- ? ABRE "TABLA
- ? PONLECTURA "TABLA
- ? PONPOSESCRITURA 0
- ? ESCRIBE "ADIOS
- ? CIERRA "TABLA
- ? HAZ "B [[] DONDE ESTAS TU [AQUI]]
- ? ESCRIBE PRIMERO :B
- ? AGRUPA "MAMIFEROS
- ? AGRUPA "EJEMPLO [A B C]
- ? IMPRESORA 1
- ? IMG "EJEMPLO
- ? PARA A
- > BL
- > REPITE 4 [AV 5 GD 90]
- > FIN
- ? PARA B
- > GD 90 AV 10
- > FIN
- ? EDG "AB

■ Solución de los ejercicios

1:

El procedimiento CARAC lee una palabra del teclado, asigna a la variable B la primera letra de la palabra, a la variable C la última letra, si la palabra tecleada es FIN detiene la ejecución, si no, se ejecuta el procedimiento ESCRIBIR y se realiza una llamada recursiva.

```
? PARA CARAC
> HAZ "A LP
> HAZ "B PRIMERO :A
> HAZ "C ULTIMO :A
> SI :A = "FIN [ALTO] [ESCRIBIR :B :C
CARAC]
> FIN
```

ESCRIBIR escribe en la pantalla la primera y la última letra de la palabra tecleada.

```
? PARA ESCRIBIR :A :B
> HAZ "C LISTA :A :B
> ESCRIBE :C
```

```
> ESCRIBE []
> FIN
```

Si tu versión de LOGO no dispone de la primitiva LP, defínete un procedimiento que la simule.

```
? PARA LEEP :D
> HAZ "X LC
> SI ASCII :X = 13 [ESCRIBE :D ALTO]
[HAZ "D PALABRA :D :X LEEP :D]
> FIN
```

Comprueba si todo funciona tecleando:
? CARAC

y las palabras que desees, por ejemplo:

```
CASA
C A
PERRO
P O
SOMBRA
S A
FIN
```

Cuando introduces la palabra FIN, la ejecución se detendrá.

2:

En este ejercicio, en vez de leer palabras, leemos listas.

```
? PARA LISTAS
> HAZ "A LL
> HAZ "B PRIMERO :A
> HAZ "C ULTIMO :A
> SI :A = [FIN] [ALTO] [ESCRIL :B :C
LISTAS]
> FIN
```

```
? PARA ESCRIL :A :B
> HAZ "C LISTA :A :B
> ESCRIBE :C
> ESCRIBE []
> FIN
```

Ejecuta el ejercicio tecleando:

```
? LISTAS
```

y las listas que desees.

3:

Para realizar el dibujo del radio-casete, definimos una serie de procedimientos que dibujan diferentes partes de éste.

Es posible saber si un objeto no contiene nada.

En primer lugar, definimos el que va a dibujar la caja:

```
? PARA CAJA
> SL
> PONPOS [-60 -40]
> BL
> REPITE 2 [AV 60 GD 90 AV 120 GD 90]
> SL
> PONPOS [-55 5]
> REPITE 2 [AV 10 GD 90 AV 110 GD 90]
> SL
> PONPOS [-22 -35]
> BL
> REPITE 2 [AV 10 GD 90 AV 44 GD 90]
> AV 33 GD 90
> AV 44 GI 90
> FIN
```

Para dibujar los altavoces definimos los procedimientos:

```
? PARA ALTAVOZ
> SL
> PONPOS [-57 -35]
> ALT
> SL
> PONPOS [22 -35]
> ALT
> FIN
```

Este procedimiento es el que sitúa a la Tortuga en el lugar adecuado para que dibuje los dos altavoces: ALT, es el que los dibuja:

```
? PARA ALT
> BL
> PONRUMBO 0
> REPITE 4 [AV 35 GD 90]
> SL
> AV 13 GD 90
> AV 5 GI 90
> BL
> REPITE 10 [AV 8 GD 36]
> SL
> GD 90 AV 2
> BL
> RELLENA
> FIN
```

La zona del casete la dibuja el siguiente procedimiento:

```
? PARA CASETE
> SL
> PONRUMBO 0
> PONPOS [-16 -21]
> BL
> REPITE 2 [AV 15 GD 90 AV 32 GD 90]
> SL
> AV 3 GD 90
> AV 6 GI 90
> BL
> REPITE 2 [AV 8 GD 90 AV 20 GD 90]
> FIN
```

Para dibujar el dial de la radio definimos DIAL:

```
? PARA DIAL
> SL
> PONPOS [-50 9]
> BL
> REPITE 2 [AV 2 GD 90 AV 100 GD 90]
> SL
> PONPOS [-30 7]
> BL
> REPITE 2 [AV 8 GD 90 AV 2 GD 90]
> SL
> AV 3 GD 90
> AV 1
> BL
> RELLENA
> SL
> GI 90 AV 2
> BL
> RELLENA
> FIN
```

El procedimiento TECLAS es el que se encarga de dibujar las teclas. Realiza dos llamadas a otros dos procedimientos que dibujan teclas diferentes en los lugares correspondientes:

```
? PARA TECLAS
> SL
> PONPOS [-23 21]
> REPITE 6 [T]
> SL
> PONPOS [-50 20]
> REPITE 2 [TT]
> FIN
```

Los grupos se pueden guardar en archivos independientemente.

EXPERIENCIA Y PRACTICAS EN LOGO

? PARA TT

- > BL
- > PONRUMBO 0
- > REPITE 2 [AV 5 GD 90 AV 3 GD 90]
- > GD 90 A 10
- > FIN

? PARA T

- > BL
- > PONRUMBO 0
- > AV 3 GD 90
- > AV 6 GD 90
- > AV 3 GI 90
- > SL
- > AV 2
- > FIN

Para dibujar el asa del radio-casete definimos:

? PARA ASA

- > SL
- > PONPOS [-60 10]
- > PONRUMBO 270
- > BL
- > AV 6 GD 90
- > AV 25 GD 90
- > AV 132 GD 90
- > AV 25 GD 90
- > AV 6
- > SL
- > PONPOS [-60 13]
- > BL
- > AV 3
- > GD 90 AV 20
- > GD 90 AV 126
- > GD 90 AV 20
- > GD 90 AV 3
- > FIN

La última parte a dibujar es la antena y lo realiza al siguiente procedimiento:

? PARA ANTENA

- > SL
- > PONPOS [55 20]
- > PONRUMBO 0
- > BL
- > REPITE 2 [AV 60 GD 90 AV 1 GD 90]
- > AV 60 GI 90 AV 1
- > REPITE 4 [GD 90 AV 3]
- > FIN

Una vez que tenemos todos los procedimientos definidos, los englobamos en uno que inicializa el proceso y los va llamando:

? PARA RADIO

- > BP
- > OT
- > CAJA
- > ALTAVOZ
- > CASETE
- > DIAL
- > TECLAS
- > ASA
- > ANTENA
- > FIN

Para ejecutarlo, bastará con introducir:

? RADIO

4:

En primer lugar, definimos el procedimiento que dibuja los 10 cuadros y el marcador superior:

? PARA CUADROS

- > SL
- > PONPOS [-80 14]
- > REPITE 10 [REPITE 4 [BL AV 8 GD 90] SL GD 90 AV 16 GI 90]
- > SL
- > PONPOS [-14 65]
- > BL
- > REPITE 2 [AV 13 GD 90 AV 20 GD 90]
- > FIN

NUM, es el que escribe los números encima de los cuadros:

? PARA NUM

- > PONCURSOR [10 9]
- > ES [0 1 2 3 4 5 6 7 8 9]
- > FIN

La flecha la va a dibujar FLECHA:

? PARA FLECHA

- > PONY -10
- > PONRUMBO 0
- > AV 15 GD 45
- > RE 5 AV 5
- > GI 90 RE 5
- > ESPERA 10
- > FIN

Las primeras órdenes de este procedimiento son el posicionamiento siempre a la misma altura con respecto del eje Y, y con rumbo 0, para que la flecha se dibuje siempre a la misma altura y con igual orientación.

Podemos editar un grupo de procedimientos.

El procedimiento CONTROL es el que se encarga, en primer lugar, de asignar un valor al azar a la variable A. Una vez realizado esto ejecuta el procedimiento VAL, que es el que va a escribir en el marcador el valor de la variable.

A continuación consulta el valor de la variable y dependiendo de cuál sea, se posiciona en un punto determinado con respecto al eje X para a continuación ir a dibujar la flecha. Una vez dibujada, vuelve a colocarse en la misma posición inicial y vuelve a dibujar la flecha, pero en esta ocasión con la goma activa, por lo que ésta se borrará.

```
? PARA CONTROL
> HAZ " AZAR 10
> VAL
> SI :A = 0 [SL PONX -76 BL FLECHA
SL PONX -76 GOMA C]
> SI :A = 1 [SL PONX -60 BL FLECHA
SL PONX -60 GOMA C]
> SI :A = 2 [SL PONX -44 BL FLECHA
SL PONX -44 GOMA C]
> SI :A = 3 [SL PONX -28 BL FLECHA
SL PONX -28 GOMA C]
> SI :A = 4 [SL PONX -12 BL FLECHA
SL PONX -12 GOMA C]
> SI :A = 5 [SL PONX 4 BL FLECHA SL
PONX 4 GOMA C]
> SI :A = 6 [SL PONX 20 BL FLECHA SL
PONX 20 GOMA C]
> SI :A = 7 [SL PONX 36 BL FLECHA SL
PONX 36 GOMA C]
> SI :A = 8 [SL PONX 52 BL FLECHA SL
PONX 52 GOMA C]
> SI :A = 9 [SL PONX 68 BL FLECHA SL
PONX 68 GOMA C]
> FIN
```

VAL escribe el valor de la variable en posición adecuada:

```
? PARA VAL
> PONCURSOR [19 5]
> ES :A
> FIN
```

Una vez que tenemos todos los procedimientos necesarios, definimos uno para que inicialice y realice todo el proceso:

```
? PARA PROCESO
> BP
```

```
> BT
> OT
> CUADROS
> NUM
> SL
> PONY -10
> REPITE 100 [CONTROL]
> FIN
```

Para ejecutarlo bastará con introducir:
? PROCESO

5:

```
- ? ABRE "TABLA
? PONLECTURA "TABLA
? CIERRA "TABLA
? ESCRIBE POSLECTURA
```

INCORRECTO

El LOGO mostrará el mensaje:
ARCHIVO NO ELEGIDO.

Estamos preguntando la posición del puntero del archivo actualmente abierto para escritura, y no hay ningún archivo abierto.

```
- ? ABRE "TABLA
? PONLECTURA "TABLA
? PONPOSESCRITURA 0
? ESCRIBE "ADIOS
? CIERRA "TABLA
```

INCORRECTO

Utilizamos la orden PONPOSESCRITURA y no hay ningún archivo abierto para escritura, el LOGO muestra el mensaje de error:

ARCHIVO NO ELEGIDO

```
- ? HAZ "B [[] DONDE ESTAS TU
[AQUI]]
? ESCRIBE PRIMERO :B
```

CORRECTO

En la pantalla aparecerá una línea en blanco, pues el primer elemento de B es la lista vacía.

```
- ? AGRUPA "MAMIFEROS
```

INCORRECTA

Falta indicar el nombre de los procedimientos que queremos que formen el grupo MAMIFEROS.

El definir grupos nos puede servir para estructurar nuestros procesos.

EXPERIENCIA Y PRACTICAS EN LOGO

— ? AGRUPA "EJEMPLO [A B C]
? IMPRESORA 1
? IMG "EJEMPLO

Agrupamos en EJEMPLO los procedimientos A, B y C.

Activamos la impresora y a continuación imprimimos el grupo. Obtendremos, tanto en la pantalla como en el papel, el listado de los procedimientos A, B y C.

— ? PARA A
> BL

> REPITE 4 [AV 5 GD 90]
> FIN

? PARA B
> GD 90 AV 10
> FIN

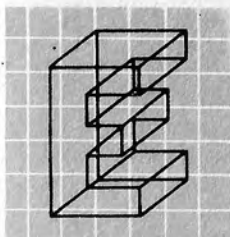
? EDG "AB

INCORRECTA

Si queremos editar el grupo AB, tendremos que haber creado anteriormente dicho grupo.

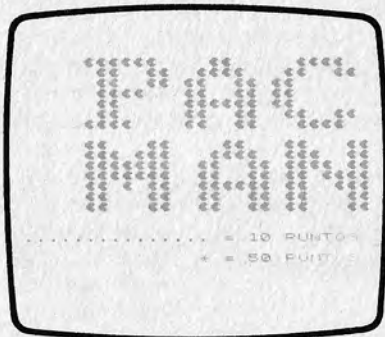
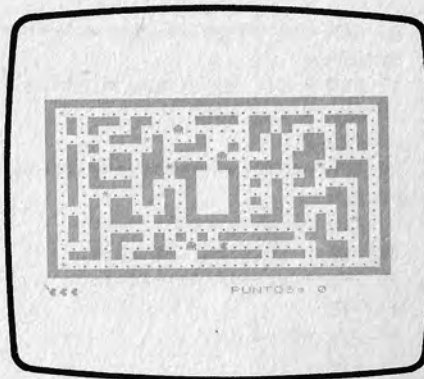
- **Puedes imprimir un grupo de procedimiento.**
- **Consulta los elementos de los objetos logo.**
- **La primitiva último se puede escribir UL.**
- **Podemos formar un grupo con todos los procedimientos que haya en la memoria.**
- **La primitiva vacío? devuelve como resultado CIERTO o FALSO.**
- **Forma grupos con procedimientos que guarden relación.**

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS



N el tomo anterior vimos el programa PAC-MAN para IBM y compatibles; en éste vamos a ver una versión de este mismo programa para el SPECTRUM.

En líneas generales, los dos programas han sido realizados de la misma manera, aunque se ha



En este caso se han utilizado dos instrucciones nuevas que no hemos estudiado. Estas son:

SCREEN (X,Y)
ATTR (X,Y)

Veamos qué función realiza cada una de ellas.

La función SCREEN (X,Y) nos devuelve el carácter que se encuentra en la columna X de la fila Y, pero no nos dice nada sobre su color. Para ello utilizamos la función ATTR (X,Y), la cual nos dice si el carácter que se encuentra en las coordenadas X,Y está parpadeando, tiene brillo normal o encendido, de qué color es su tinta y su papel.

El resultado de ATTR es un número entre 0 y 255. En este número (que es un BYTE) cada BIT tiene un valor específico y nos informa de una parte del atributo.

- Los tres primeros BITS me dicen el color de la tinta.
- Los segundos tres BITS me dicen el color del papel.

intentado sacar el mayor partido posible a las características específicas de cada ordenador.

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

— El BIT número 6 me dice si el carácter tiene brillo normal (0) o resaltado (1).

— El último BIT (el séptimo) me dice si dicho carácter está en FLASH (1) o normal (0).

```

10 REM *****
11 REM * *
12 REM * P A C - M A N *
13 REM * *
14 REM *****
15 REM
16 REM *****
17 REM * *
18 REM * PROGRAMADO POR: *
19 REM * *
20 REM * Fco. Morales Guerrero. *
21 REM * *
22 REM *****
23 REM
24 REM *****
25 REM * *
26 REM * (c) Ediciones Siglo Cultural *
27 REM * (c) 1987. *
28 REM * *
29 REM *****
30 REM
31 REM *****
32 REM * INICIALIZACION *
33 REM *****
34 REM
40 DIM N$(10,14)
50 DIM P(10)
55 LET PU=0
60 BORDER 0
70 PAPER 0
80 INK 7
90 CLS
100 PRINT AT 0,8; INK 5;"P A C - M A N"
110 PRINT AT 1,7; INK 3;"-----"
120 PRINT AT 10,9; INK 6; FLASH 1;"CARGANDO DATAS"
130 PRINT AT 12,7; INK 4; INVERSE 1; FLASH 1;"ESPERE UN MOMENTO."
140 GO SUB 7000
175 GO SUB 1870
180 PRINT #0:AT 0,8; FLASH 1;"PULSA UNA TECLA."
190 IF INKEY$="" THEN GO TO 190
192 GO SUB 3000
194 REM
195 REM *****
196 REM * PROGRAMA PRINCIPAL *
197 REM *****
198 REM
200 CLS
210 LET VI=3
220 LET V$=CHR$ 144+CHR$ 144+CHR$ 144
230 LET PU=0
240 LET PA=1
241 REM
242 REM *****
243 REM * LECTURA DE LAS DATAS DE LA PANTALLA *
244 REM *****
245 REM
260 RESTORE 310

```

```

270 FOR F=0 TO 19
280   READ a$
290   FOR J=1 TO LEN (A$)
291     IF A$(J)=" " THEN PRINT " ";
292     IF A$(J)="0" THEN PRINT INK 3: PAPER 3: "+";
293     IF A$(J)="1" THEN PRINT INK 7; ".";
294     IF A$(J)="2" THEN PRINT "*";
295   NEXT J
296   PRINT
300 NEXT F
301 REM
302 REM *****
303 REM * LINEAS DATA CON LA DEFINICION DE LA PANTALLA *
304 REM *****
305 REM
310 DATA "00000000000000000000000000000000"
320 DATA "02111111111111111111111111111120"
330 DATA "010100010010010000010010001000010"
340 DATA "010101110111011111110010111101010"
350 DATA "01111100020111010110101100101010"
360 DATA "0101011110100010010101100101010"
370 DATA "01011100100111111111101000111110"
380 DATA "010101001002000 0010001002100010"
390 DATA "01010111100100 011111101000010"
400 DATA "01010000101110 010101001111010"
410 DATA "01011211101010 011101111001010"
420 DATA "01010100101010 010101000001010"
430 DATA "01110100111010 011101011101010"
440 DATA "01010100010010000010001010102010"
450 DATA "010101111111111111111111110101010"
460 DATA "01010100010101010000000100111010"
470 DATA "011101011101111111111111110001010"
480 DATA "010001010000000000000100010001010"
490 DATA "02111111111111111111111111111120"
500 DATA "00000000000000000000000000000000"
505 REM
510 PRINT AT 21,17: INK 2;"PUNTOS="; INK 6;PU
512 PRINT AT 21,0: INK 6;V$
515 LET CC=0
516 LET S1=0
517 LET S2=0
518 LET S3=0
520 LET X=16
530 LET Y=16
540 LET E$=""
550 LET W$=""
560 LET Q$=""
565 LET P$=""
570 LET X1=10
580 LET Y1=1
590 LET X2=15
600 LET Y2=4
610 LET X3=11
620 LET Y3=18
630 LET X4=1
640 LET Y4=18
650 PRINT AT Y,X:CHR$ 144
660 POKE 23658,8
670 LET K$=INKEY$
671 IF K$="" THEN LET K$=P$
672 LET P$=K$
680 IF K$="Q" THEN IF SCREEN$ (Y,X-1)<>"+" THEN LET X=X-1: PRINT AT Y,X+1;" "

```

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

```

: LET A$=SCREEN$ (Y,X): PRINT AT Y,X:CHR$ 145: GO TO 712
690 IF K$="W" THEN IF SCREEN$ (Y,X+1)<>"+" THEN LET X=X+1: PRINT AT Y,X-1;" "
: LET A$=SCREEN$ (Y,X): PRINT AT Y,X:CHR$ 144: GO TO 712
700 IF K$="O" THEN IF SCREEN$ (Y-1,X)<>"+" THEN LET Y=Y-1: PRINT AT Y+1,X;" "
: LET A$=SCREEN$ (Y,X): PRINT AT Y,X:CHR$ 146: GO TO 712
710 IF K$="K" THEN IF SCREEN$ (Y+1,X)<>"+" THEN LET Y=Y+1: PRINT AT Y-1,X;" "
: LET A$=SCREEN$ (Y,X): PRINT AT Y,X:CHR$ 147
712 IF A$="." THEN LET PU=PU+10: IF CC<>0 THEN LET CC=CC-1
714 IF A$="*" THEN LET PU=PU+50: LET CC=25*VI
715 LET A$=""
720 IF INT (PU/PA)=3430 THEN GO TO 1170
730 GO SUB 800
740 GO SUB 870
750 GO SUB 950
760 IF ATTR (Y,X)<>7 THEN GO SUB 1030
770 PRINT AT 21,25: INK 6:PU
780 GO TO 680
785 REM
790 REM *****
800 REM * MOVIMIENTO DEL MARCIANO 1 *
810 REM *****
815 REM
816 IF CC<>0 AND S1=1 THEN LET S1=0: RETURN
817 LET S1=1
820 LET I1=SGN (X-X1)
830 LET A1=SGN (Y-Y1)
840 IF SCREEN$ (Y1,X1+I1)<>"+" THEN LET X1=X1+I1: PRINT AT Y1,X1-I1;Q$: LET Q$
=SCREEN$ (Y1,X1): PRINT AT Y1,X1: INK 1:CHR$ 148
850 IF SCREEN$ (Y1+A1,X1)<>"+" THEN LET Y1=Y1+A1: PRINT AT Y1-A1,X1;Q$: LET Q$
=SCREEN$ (Y1,X1): PRINT AT Y1,X1: INK 1:CHR$ 148
860 RETURN
865 REM
870 REM *****
880 REM * MOVIMIENTO DEL MARCIANO 2 *
890 REM *****
895 REM
896 IF CC<>0 AND S2=1 THEN LET S2=0: RETURN
897 LET S2=1
900 LET I2=SGN (X-X2)
910 LET A2=SGN (Y-Y2)
920 IF SCREEN$ (Y2,X2+I2)<>"+" THEN LET X2=X2+I2: PRINT AT Y2,X2-I2;W$: LET W$
=SCREEN$ (Y2,X2): PRINT AT Y2,X2: INK 2:CHR$ 148
930 IF SCREEN$ (Y2+A2,X2)<>"+" THEN LET Y2=Y2+A2: PRINT AT Y2-A2,X2;W$: LET W$
=SCREEN$ (Y2,X2): PRINT AT Y2,X2: INK 2:CHR$ 148
940 RETURN
945 REM
950 REM *****
960 REM * MOVIMIENTO DEL MARCIANO 3 *
970 REM *****
975 REM
976 IF CC<>0 AND S3=1 THEN LET S3=0: RETURN
977 LET S3=1
980 LET I3=SGN (X-X3)
990 LET A3=SGN (Y-Y3)
1000 IF SCREEN$ (Y3,X3+I3)<>"+" THEN LET X3=X3+I3: PRINT AT Y3,X3-I3;E$: LET E$
=SCREEN$ (Y3,X3): PRINT AT Y3,X3: INK 4:CHR$ 148
1010 IF SCREEN$ (Y3+A3,X3)<>"+" THEN LET Y3=Y3+A3: PRINT AT Y3-A3,X3;E$: LET E$
=SCREEN$ (Y3,X3): PRINT AT Y3,X3: INK 4:CHR$ 148
1020 RETURN
1025 REM
1030 REM *****
1040 REM * PERDIDA DE UNA VIDA *
1050 REM *****

```

```

1055 REM
1056 FOR I=0 TO 10
1057   PRINT AT Y,X:CHR$ 144
1058   BEEP .2,-10
1059   PRINT AT Y,X;" "
1060   BEEP .2,-20
1061 NEXT I
1062 PRINT AT Y1,X1:Q$
1063 PRINT AT Y2,X2:W$
1064 PRINT AT Y3,X3:E$
1065 PRINT AT 21,0: FLASH 1;"      * * M U E R T O * *      "
1070 FOR F=30 TO 0 STEP -1
1080   BEEP .1,F
1090 NEXT F
1100 LET VI=VI-1
1110 IF VI<0 THEN GO TO 1750
1120 LET V$=V$( TO VI)
1130 PRINT #0;AT 1,0; INK 7; PAPER 1; FLASH 1;"      PULSA UNA TECLA
1140 IF INKEY$="" THEN GO TO 1140
1150 INPUT ""
1151 PRINT AT 21,0: INK 4;"
1160 GO TO 510
1165 REM
1170 REM *****
1180 REM * TERMINACION DE UNA PANTALLA *
1190 REM *****
1195 REM
1201 FOR I=1 TO 15
1202   FOR J=0 TO 7
1203     BORDER J
1204   NEXT J
1205 NEXT I
1206 BORDER 0
1207 CLS
1208 LET PA=PA+1
1210 LET B=31
1220 LET A$=CHR$ 16+CHR$ 2+CHR$ 148+" "+CHR$ 16+CHR$ 4+CHR$ 148+" "+CHR$ 16+CHR$
3+CHR$ 148+" "
1230 FOR F=26 TO 3 STEP -1
1240   PRINT INK 1;AT 10,F:A$
1250   IF F<18 THEN PRINT AT 10,B:CHR$ 145+" ": LET B=B-2
1260   IF B<7 THEN LET A$=A$( TO LEN A$-4)
1270   FOR X=1 TO 5
1275     NEXT X
1280 NEXT F
1290 FOR F=3 TO 27
1300   PRINT AT 10,F; INK 6;" "+CHR$ 144
1310   BEEP .005,30
1320 NEXT F
1330 FOR F=10 TO 15
1340   PRINT AT F,28; INK 6;CHR$ 147;AT F-1,28;" "
1350   BEEP .005,50
1360 NEXT F
1370 LET A=144
1380 FOR F=1 TO 21
1390   PRINT AT 15,28; INK 6;CHR$ A
1400   LET A=A+1
1410   IF A=148 THEN LET A=144
1420   FOR X=1 TO 4
1430     NEXT X
1440   BEEP .05,-50+INT (RND*100)
1450 NEXT F

```

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

```

1460 PRINT AT 3,0;" Has conseguido comer los 306"
1461 PRINT AT 4,0;"huevecillos de los celosos fan-"
1462 PRINT AT 5,0;"tasmás de las oscuras catacumbas"
1463 PRINT AT 6,0;"envejecidas por el monotonó rras"
1464 PRINT AT 7,0;"trec de sus omisas extremidades."
1470 PRINT AT 13,10: FLASH 1:"BONOS ";VI*1000
1480 PRINT AT 15,0;"Puntuacion ";PU;"+";VI*1000;"=";PU+VI*1000
1490 PRINT AT 21,9: FLASH 1;"PULSA UNA TECLA"
1500 IF INKEY$="" THEN GO TO 1500
1510 FOR F=15 TO 0 STEP -1
1520   PRINT AT F,28; INK 6;CHR$ 146;AT F+1,28;" "
1530 NEXT F
1540 FOR F=28 TO 0 STEP -1
1550   PRINT AT 0,F; INK 6;CHR$ 145+" "
1560 NEXT F
1570 PRINT AT 0,0;" ";AT 1,0; INK 6;CHR$ 144
1590 LET A=7
1595 LET B=3
1600 PRINT AT 1,0;" "
1610 FOR W=1 TO 2
1620   FOR C=B TO A
1630     FOR F=0 TO 30
1640       PRINT AT C,F; INK 6;" "+CHR$ 144
1650       NEXT F
1660       PRINT AT C,31;" "
1670     NEXT C
1680   LET A=21
1690   LET B=13
1700 NEXT W
1710 CLS
1720 LET PU=PU+VI*1000
1740 GO TO 260
1745 REM
1750 REM *****
1760 REM * FINAL DE LA PARTIDA *
1770 REM *****
1775 REM
1780 CLS
1790 PRINT AT 10,2;"Tu puntuacion ha sido=";PU
1800 PRINT " " Lastima,parecias bueno."
1810 PRINT " No te preocupes.cuenta con mi compasion."
1820 PRINT #0: FLASH 1:"PULSA UNA TECLA PARA VER RECORDS"
1830 LET K$=INKEY$
1840 IF K$="" THEN GO TO 1830
1850 GO SUB 4000
1860 GO TO 180
1865 REM
1870 REM *****
1880 REM * LECTURA DE LAS DATA CON LA DEFINICION DE LOS CARACTERES *
1890 REM *****
1895 REM
1900 RESTORE 1980
1910 FOR F=97 TO 101
1920   FOR C=0 TO 7
1930     READ A
1940     POKE USR (CHR$ F)+C,A
1950   NEXT C
1960 NEXT F
1970 RETURN
1971 REM
1972 REM *****
1973 REM * DATAS CON LA DEFINICION DE LOS CARACTERES *
1974 REM *****

```

```

1975 REM
1980 DATA 0,28,62,124,120,124,62,28
1990 DATA 0,56,124,62,30,62,124,56
2000 DATA 0,34,119,127,127,62,28,0
2010 DATA 0,56,124,254,254,238,68,0
2020 DATA 56,124,214,254,238,254,170,170
3000 REM
3010 REM *****
3020 REM * PRESENTACION *
3030 REM *****
3040 REM
3050 CLS
3060 RESTORE 9000
3070 FOR I=0 TO 14
3080     PRINT AT I,5;
3090     READ A$
3100     FOR J=1 TO LEN (A$)
3110         IF A$(J)="1" THEN PRINT CHR$ (144);
3120         IF A$(J)="0" THEN PRINT " ";
3130     NEXT J
3140 NEXT I
3150 FOR I=0 TO 14
3160     PRINT AT 17,I;". "
3170     PAUSE 4
3180 NEXT I
3190 PRINT AT 17,14;". = 10 PUNTOS"
3200 FOR I=0 TO 14
3210     PRINT AT 19,I;"*"
3220     PAUSE 7
3230     PRINT AT 19,I;" "
3250 NEXT I
3260 PRINT AT 19,14;"* = 50 PUNTOS"
3270 FOR I=1 TO 150
3280 NEXT I
3290 FOR I=0 TO 14
3300     PRINT AT 17,I:CHR$ 144
3310     PAUSE 7
3320 PRINT AT 17,I;" "
3330 NEXT I
3340 PRINT AT 17,14:CHR$ 144;" = TU MISMO "
3350 FOR I=0 TO 14
3360     PRINT AT 19,I:CHR$ 148
3370     PAUSE 7
3380     PRINT AT 19,I;" "
3390 NEXT I
3400 PRINT AT 19,14:CHR$ 148;" = TU ENEMIGO"
3410 FOR I=1 TO 150
3420 NEXT I
3430 GO SUB 4000
3440 RETURN
4000 REM
4010 REM *****
4020 REM * TABLA DE RECORDS *
4030 REM *****
4040 REM
4045 CLS
4050 PRINT INK 5;AT 0,8;"TABLA DE RECORDS"
4060 PRINT INK 6;AT 1,7;"-----"
4070 PRINT AT 3,0;" ";
4080 FOR I=0 TO 27
4090     PRINT CHR$ 143;
4100 NEXT I
4110 PRINT

```

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

```

4120 PRINT " ";CHR$ 143:TAB 29:CHR$ 143
4130 PRINT " ";
4140 FOR I=0 TO 27
4150   PRINT CHR$ 143:
4160 NEXT I
4170 PRINT
4180 FOR I=1 TO 9
4190   PRINT " ";CHR$ 143:TAB 29:CHR$ 143
4200 NEXT I
4210 PRINT " ";
4220 FOR I=0 TO 27
4230   PRINT CHR$ 143:
4240 NEXT I
4250 LET NP=0
4260 FOR I=1 TO 10
4270   IF PU>P(I) THEN LET NP=1
4280 NEXT I
4290 IF NP=1 THEN LET P(10)=PU: LET N$(10,1)=CHR$ 255
4300 FOR I=1 TO 10
4310   FOR J=1 TO I
4320     IF P(I)>P(J) THEN LET A$=N$(I): LET A=P(I): LET N$(I)=N$(J): LET P(I)
4330     =P(J): LET N$(J)=A$: LET P(J)=A
4340   NEXT J
4350 IF NP=0 THEN GO TO 4390
4360 FOR I=1 TO 10
4370   IF N$(I,1)=CHR$ 255 THEN LET N$(I)="-": LET NP=1
4380 NEXT I
4390 PRINT AT 4,4: FLASH 1: INK 1: PAPER 8:N$(1);" ... ";P(1)
4400 FOR I=2 TO 10
4410   PRINT INK 4:AT 4+I,4:N$(I);" ... ";P(I)
4420 NEXT I
4430 IF NP=0 THEN GO TO 4580
4435 INK 4: IF NP=1 THEN FLASH 1: INK 6: PAPER 1
4440 PRINT AT 17,11: FLASH 1: INK 5;"FELICIDADES"
4450 PRINT AT 19,3:"TU PUNTUACION ES MUY BUENA"
4460 PRINT AT 21,7: INK 4;"ESCRIBE TU NOMBRE.";
4470 LET N$(NP)=""
4480 LET Y=4: IF NP<>1 THEN LET Y=4+NP
4490 PRINT AT Y,X;
4500 PRINT " ";CHR$ 8;
4510 PAUSE 6: LET A$=INKEY$: IF A$="" THEN GO TO 4510
4520 IF A$=CHR$ 12 AND LO<>1 THEN LET LO=LO-1: LET N$(NP,LO)="":: PRINT CHR$ 8
4530 IF A$="" THEN GO TO 4560
4540 IF A$=CHR$ 13 THEN GO TO 4570
4550 IF A$<"0" OR A$>"z" THEN GO TO 4510
4560 PRINT A$;" ";CHR$ 8:: LET N$(NP,LO)=A$: LET LO=LO+1: IF LO<>15 THEN GO TO
4570 PRINT AT Y,4:N$(NP)
4580 PRINT #0;" PULSA UNA TECLA PARA CONTINUAR"
4590 PAUSE 0
4600 PAPER 0: INK 7: FLASH 0
4610 RETURN
4999 STOP
7000 REM
7010 REM *****
7020 REM * LECTURA DE LOS RECORDS *
7030 REM *****
7040 REM
7050 RESTORE 8000
7060 FOR I=1 TO 10

```

```

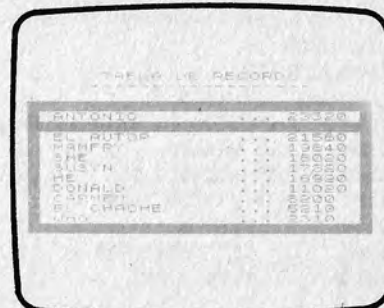
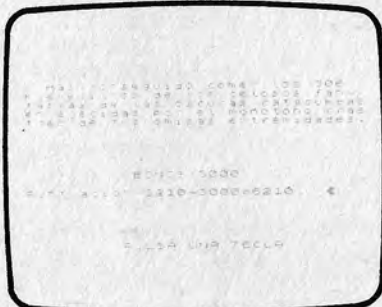
7070 READ N$(I)
7080 READ P(I)
7090 NEXT I
7100 RETURN
8000 REM
8010 REM *****
8020 REM * LINEAS DATA CON LOS RECORDS *
8030 REM *****
8040 REM
8050 DATA "ANTONIO",23320
8060 DATA "EL AUTOR",21560
8070 DATA "HAMFRY",19840
8080 DATA "SHE",18020
8090 DATA "HE",16920
8100 DATA "SUSYN",17320
8110 DATA "DONALD",11020
8120 DATA "CARMEN",8200
8130 DATA "EL CHACHE",5210
8140 DATA "WHO",2310
9000 REM
9010 REM *****
9020 REM * DATAS CON LA PALABRA PAC-MAN *
9030 REM *****
9040 REM
9050 DATA "11111100001100000111110"
9060 DATA "0110011001111000110011"
9070 DATA "0110011011001101100000"
9080 DATA "0111100011001101100000"
9090 DATA "0110000011111101100000"
9100 DATA "0110000011001100110011"
9110 DATA "11110000110011000111110"
9120 DATA ""
9130 DATA "11000110000110001100011"
9140 DATA "11101110001111001110011"
9150 DATA "11111110011001101111011"
9160 DATA "11111110011001101101111"
9170 DATA "11010110011111101100111"
9180 DATA "11000110011001101100011"
9190 DATA "11000110011001101100011"

```

Las teclas que tendremos que utilizar para mover nuestro quesito por la pantalla son las siguientes:

- Q: Movimiento hacia la izquierda.
- W: Movimiento hacia la derecha.
- O: Movimiento hacia arriba.
- K: Movimiento hacia abajo.

Como ejercicio al lector, se propone crear una rutina que nos permita redefinir las teclas de movimiento a gusto del usuario al principio del programa.



TRUCOS Y RUTINAS BASICAS



IGUIENDO con las rutinas específicas para cada ordenador que nos permitirán sacarle más provecho a nuestros programas vamos a ver a continuación una rutina que nos permitirá hacer SCROLL de los atributos del

SPECTRUM en las cuatro direcciones de la pantalla.

```

10 REM *****
20 REM *
30 REM * SCROLL DE ATRIBUTOS *
40 REM *
50 REM * PROGRAMA DEMO *
60 REM * ----- *
70 REM *
80 REM *****
90 REM
100 REM *****
110 REM *
120 REM * (c) Ediciones Siglo Veintiuno *
130 REM *
140 REM * (c) 1987 *
150 REM *
160 REM *****
170 REM
1000 PAPER 0
1010 BORDER 0
1020 INK 6
1030 CLEAR 54999
1060 PRINT AT 10,10; FLASH 1;"CARGANDO DATAS"
1070 PRINT AT 12,8; FLASH 1; INVERSE 1; INK 4;"ESPERE UN MOMENTO."
1080 REM
1090 REM *****
1100 REM * LECTURA DEL CODIGO MAQUINA *
1110 REM *****
1120 REM
1130 LET TOT=0
1140 LET LIN=9040
1150 FOR I=55000 TO 55164 STEP 15
1160 LET CHEK=0
1170 LET LIN=LIN+10
1180 FOR J=I TO I+14
1190 READ A
1200 LET CHEK=CHEK+A
1210 POKE J,A

```

```

1220     NEXT J
1230     READ TOT
1240     IF TOT<>CHEK THEN GO TO 5000
1250 NEXT I
2000 REM
2010 REM *****
2020 REM *
2030 REM * DEMOSTRACION *
2040 REM *
2050 REM *****
2060 REM
2070 CLS
2080 OVER 1
2090 FOR I=0 TO 7
2100     PAPER I
2110     FOR J=0 TO 21
2120         PRINT AT J,I*3+2;" "
2130     NEXT J
2140 NEXT I
2150 OVER 0
2160 PAUSE 50
2170 REM
2180 REM *****
2190 REM * SCROLL A LA IZQUIERDA *
2200 REM *****
2210 REM
2220 POKE 23300,1 : REM SCROLL A LA IZQUIERDA
2230 POKE 23301,7 : REM NUMERO DE LINEAS. DESDE LA 0 A LA 6
2240 POKE 23302,7 : REM COLOR QUE APARECE EN LA PARTE DERECHA
2250 REM
2260 FOR I=0 TO 31
2270     RANDOMIZE USR 55000
2280     PAUSE 4
2290 NEXT I
2295 PAUSE 50
2300 REM
2310 REM *****
2320 REM * SCROLL A LA DERECHA *
2330 REM *****
2340 REM
2350 POKE 23300,2 : REM SCROLL A LA DERECHA
2360 POKE 23301,12: REM NUMERO DE LINEAS. DESDE LA 0 A LA 9
2370 POKE 23302,7 : REM COLOR QUE APARECE EN LA PARTE IZQUIERDA
2380 REM
2390 FOR I=0 TO 31
2400     RANDOMIZE USR 55000
2410     PAUSE 4
2420 NEXT I
2430 PAUSE 50
2440 REM
2450 REM *****
2460 REM * SCROLL HACIA ARRIBA *
2470 REM *****
2480 REM
2490 POKE 23300,3 : REM SCROLL HACIA ARRIBA
2500 POKE 23301,21: REM NUMERO DE LINEAS. DESDE LA 0 A LA 19
2510 POKE 23302,7 : REM COLOR QUE APARECE EN LA PARTE INFERIOR
2520 REM
2530 FOR I=0 TO 10
2540     RANDOMIZE USR 55000
2550     PAUSE 4
2560 NEXT I
2570 PAUSE 50
2580 REM
2590 REM *****
2600 REM * SCROLL HACIA ABAJO *

```

TRUCOS Y RUTINAS BASICAS

```

2610 REM *****
2620 REM
2630 POKE 23300,4 : REM SCROLL HACIA ABAJO
2640 POKE 23301,21: REM NUMERO DE LINEAS. DESDE LA 0 A LA 19
2650 POKE 23302,7 : REM COLOR QUE APARECE EN LA PARTE SUPERIOR
2660 REM
2670 FOR I=0 TO 10
2680     RANDOMIZE USR 55000
2690     PAUSE 4
2700 NEXT I
2710 PAUSE 50
2720 REM
2730 REM *** REPETICION **
2735 PAPER 0
2740 REM
2750 CLS
2760 FOR I=0 TO 20
2770     FOR J=0 TO I
2780         PRINT " ";
2790     NEXT J
2800     PRINT "SCROLL"
2810 NEXT I
2820 PRINT AT 0,0;
2830 GO TO 2080
4999 PAPER 0: STOP
5000 REM
5010 REM *****
5020 REM * ERROR EN LAS LINEAS DATA *
5030 REM *****
5040 REM
5050 CLS
5060 PRINT "HA HABIDO UN ERROR EN LAS DATA"
5070 PRINT : PRINT
5080 PRINT "EL ERROR ESTA EN LA LINEA ";LIN
5090 PRINT : PRINT
5100 PRINT "REPASA DICHA LINEA"
5110 PRINT : PRINT
5120 LIST LIN
5130 STOP
9000 REM
9010 REM *****
9020 REM * LINEAS DATA CON EL CODIGO MAQUINA *
9030 REM *****
9040 REM
9050 DATA 0,58,4,91,254,1,40,13,254,2,40,34,254,3,40
9055 DATA 1088
9060 DATA 62,254,4,40,91,201,33,0,88,58,5,91,79,58,6
9065 DATA 1070
9070 DATA 91,6,31,35,94,43,115,35,16,249,119,35,13,32,242
9075 DATA 1156
9080 DATA 201,33,255,87,17,32,0,58,5,91,71,79,25,16,253
9085 DATA 1223
9090 DATA 58,6,91,6,31,43,94,35,115,43,16,249,119,43,13
9095 DATA 962
9100 DATA 32,242,201,33,0,0,17,32,0,58,5,91,71,25,16
9105 DATA 823
9110 DATA 253,68,77,33,32,88,17,0,88,237,176,58,6,91,6
9115 DATA 1230
9120 DATA 32,18,19,16,252,201,33,0,0,17,32,0,58,5,91
9125 DATA 774
9130 DATA 71,25,16,253,68,77,197,33,0,88,71,25,16,253,229
9135 DATA 1422
9140 DATA 25,235,225,193,237,184,58,6,91,6,32,17,0,88,18
9145 DATA 1415
9150 DATA 19,16,252,201,0,0,0,0,0,0,0,0,0,0,0
9155 DATA 488

```

Este programa es sólo de demostración para poder ver cómo funciona la rutina; el programa 2 es el código fuente de dicha rutina.

```

1000 :*****
1010 :*
1020 :*  RUTINAS  DE  SCROLL  *
1030 :*  DE  LOS  ATRIBUTOS  *
1040 :*
1050 :*****
1060 :
1070 :
1080 :*****
1090 :*
1100 :* (C) EDICIONES SIGLO CULTURAL *
1110 :*
1120 :* (C) 1987
1130 :*
1140 :*****
1150 :
1160 :
1170 :=====
1180 :==
1190 :== COMO EL PROGRAMA PUEDE REALIZAR SCROLL ==
1200 :== EN LAS CUATRO DIRECCIONES DE LA PANTALLA ==
1210 :== TENEMOS QUE ALMACENAR EN LA POSICION DE ==
1220 :== MEMORIA 23301 LA DIRECCION QUE QUEREMOS ==
1230 :== QUE TENGA EL SCROLL SEGUN LA TABLA QUE ==
1240 :== APARECE A CONTINUACION:
1250 :==
1260 :== 1 = SCROLL A LA IZQUIERDA
1270 :== 2 = SCROLL A LA DERECHA
1280 :== 3 = SCROLL HACIA ARRIBA
1290 :== 4 = SCROLL HACIA ABAJO
1300 :==
1310 :=====
1320 :
1330 :
1340 DIR EQU 23300
1350 :
1360 :
1370 :=====
1380 :==
1390 :== EL NUMERO DE LINEAS QUE QUEREMOS MOVER ==
1400 :== CADA VEZ QUE SE EJECUTE LA RUTINA TENE- ==
1410 :== MOS QUE ALMACENARLAS EN LA DIRECCION DE ==
1420 :== MEMORIA 23301.
1430 :== DICHO NUMERO ESTARA COMPRENDIDO ENTRE ==
1440 :== 1 Y 24 AMBOS INCLUSIVE.
1450 :==
1460 :=====
1470 :
1480 :
1490 NUM EQU 23301
1500 :
1510 :
1520 :=====
1530 :==
1540 :== EL ATRIBUTO QUE TENDRA LA LINEA QUE SE ==
1550 :== QUEDA VACIA TRAS EL SCROLL A DE IR ALMA ==
1560 :== CENADO EN LA DIRECCION 23302.
1570 :== EL ATRIBUTO SERA UN NUMERO ENTRE 0 Y =

```

TRUCOS Y RUTINAS BASICAS

```

1580 := 255!
1590 :=
1600 :=
1610 :
1620 :
1630 ATTR EQU 23302
1640 :
1650 :
1660 :=
1670 :=
1680 := ESTE PROGRAMA ES TOTALMENTE REUBICABLE, =
1690 := POR ELLO, AUNQUE EL PROGRAMA SE REALIZO =
1700 := A PARTIR DE LA DIRECCION 55000, EL USUA- =
1710 := RIO PUEDE ELEGIR LA DIRECCION QUE MAS LE =
1720 := CONVENGA PARA SU PROGRAMA.
1730 :=
1740 :=
1750 :
1760 :
1770 ORG 55000
1780 :
1790 :
1800 :*****
1810 :*****
1820 :***** PROGRAMA PRINCIPAL *****
1830 :*****
1840 :*****
1850 :
1860 :
1870 LD A,(DIR) ; ALMACENAMOS LA DIRECCION DEL SCROLL
1880 :
1890 CP 1
1900 JR Z,IZQ ; SI ES 1 VAMOS A 'IZQ'
1910 :
1920 CP 2
1930 JR Z,DER ; SI ES 2 VAMOS A 'DER'
1940 :
1950 CP 3
1960 JR Z,ARR ; SI ES 3 VAMOS A 'ARR'
1970 :
1980 CP 4
1990 JR Z,ABB ; SI ES 4 VAMOS A 'ABB'
2000 :
2010 RET ; SI ES OTRO NUMERO VOLVEMOS AL BASIC
2020 :
2030 :
2040 :*****
2050 :*
2060 :* SCROLL A LA IZQUIERDA *
2070 :*
2080 :*****
2090 :
2100 :
2110 IZQ LD HL,22528 ; HL = AREA DE ATRIBUTOS
2120 LD A,(NUM) ; A = NUMERO DE LINEAS A MOVER
2130 LD C,A ; C = A
2140 LD A,(ATTR) ; A = COLOR DE LA LINEA VACIA
2150 :
2160 LIN-IZ LD B,31 ; B = NUMERO DE CHR POR LINEA
2170 :
2180 CHR-IZ INC HL ; HL = HL + 1
2190 LD E,(HL) ; E = CONTENIDO DE HL

```

```

2200      DEC      HL      ; HL = HL - 1
2210      LD      (HL),D  ; HL = CONTENIDO DE LA POSICION ANT.
2220      INC      HL      ; HL = HL + 1
2230      DJNZ     CHR-IZ  ; SI NO HEMOS TERMINADO CONTINUAMOS
2240 ;
2250      LD      (HL),A   ; PONEMOS EL ATRIBUTO
2260      INC      HL      ; HL = HL + 1
2270      DEC      C       ; UNA LINEA MENOS. C = C - 1
2280      JR       NZ,LIN-IZ ; CONTINUAMOS SI QUEDAN LINEAS
2290 ;
2300      RET                      ; Y SI NO VOLVEMOS AL BASIC.
2310 ;
2320 ;
2330 ;*****
2340 ;*                               *
2350 ;*  SCROLL A LA DERECHA  *
2360 ;*                               *
2370 ;*****
2380 ;
2390 ;
2400 DER      LD      HL,22527 ; HL = ZONA DE ATRIBUTOS - 1
2410      LD      DE,32        ; DE = NUMERO DE BYTES POR LINEA
2420      LD      A,(NUM)      ; A = NUMERO DE LINEAS
2430      LD      B,A          ; B = A
2440      LD      C,A          ; C = A
2450 ;
2460 SUM-DE    ADD     HL,DE    ; HL = HL + DE
2470      DJNZ     SUM-DE      ; B = B - 1! IF B<>0 THEN SUM-DE
2480      ;                  ; B = DIRECCION DE LA ULTIMA LINEA
2490 ;
2500      LD      A,(ATTR)    ; A = ATRIBUTO A INTRODUCIR
2510 ;
2520 LIN-DE    LD      B,31    ; B = NUMERO DE CHR POR LINEA - 1
2530 ;
2540 CHR-DE    DEC     HL      ; HL = HL - 1
2550      LD      E,(HL)      ; E = LO QUE HAY EN HL
2560      INC     HL          ; HL = HL + 1
2570      LD      (HL),E      ; GUARDAMOS LO QUE HABIA EN E
2580      DEC     HL          ; HL = HL - 1
2590      DJNZ     CHR-DE     ; B = B - 1: IF B<>0 THEN CHR-DE
2600 ;
2610      LD      (HL),A       ; PONEMOS EL ATRIBUTO
2620      DEC     HL          ; HL = HL - 1
2630      DEC     C           ; C = C - 1
2640      JR       NZ,LIN-DE  ; IF C<>0 THEN LIN-DE
2650 ;
2660      RET                      ; VOLVEMOS AL BASIC.
2670 ;
2680 ;
2690 ;*****
2700 ;*                               *
2710 ;*  SCROLL HACIA ARRIBA  *
2720 ;*                               *
2730 ;*****
2740 ;
2750 ;
2760 ARR      LD      HL,0      ; HL = 0
2770      LD      DE,32        ; DE = NUMERO DE CHR POR LINEA
2780      LD      A,(NUM)      ; A = NUMERO DE LINEAS A MOVER
2790      LD      B,A          ; B = A
2800 ;
2810 SUM-AR    ADD     HL,DE    ; HL = HL + DE
2820      DJNZ     SUM-AR      ; B = B - 1: IF B<>0 THEN SUM-AR

```

TRUCOS Y RUTINAS BASICAS

```

2830 ; HL = NUMERO DE BYTES A MOVER
2840 ;
2850 LD B,H
2860 LD C,L ; BC = HL. BC = NUMERO DE BYTES A MOVER
2870 LD HL,22560 ; HL = ORIGEN PARA LDIR
2880 LD DE,22528 ; DE = DESTINO PARA LDIR
2890 LDIR ; LD (DE), (HL):HL=HL+1:DE=DE+1:BC=BC-1
2900 ;
2910 LD A,(ATTR) ; A = ATRIBUTO A INTRODUCIR
2920 LD B,32 ; B = CARACTERES POR LINEA
2930 ;
2940 CHR-AR LD (DE),A ; PONEMOS EL ATRIBUTO
2950 INC DE ; DE = DE + 1
2960 DJNZ CHR-AR ; B = B - 1:IF B<>0 THEN CHR-ARR
2970 ;
2980 RET ; VUELTA AL BASIC
2990 ;
3000 ;
3010 ;*****
3020 ;* *
3030 ;* SCROLL HACIA ABAJO *
3040 ;* *
3050 ;*****
3060 ;
3070 ;
3080 ABB LD HL,0 ; HL = 0
3090 LD DE,32 ; DE = NUMERO DE CHR POR LINEA
3100 LD A,(LIN) ; A = NUMERO DE LINEAS A MOVER
3110 LD B,A ; B = A
3120 ;
3130 SU1-AB ADD HL,DE ; HL = HL + DE
3140 DJNZ SU1-AB ; B = B - 1:IF B<>0 THEN SU1-AB
3150 ; HL = NUMERO DE BYTES A MOVER
3160 ;
3170 LD B,H ; B = H
3180 LD C,L ; C = L
3190 ; BC = NUMERO DE BYTES A MOVER
3200 PUSH BC ; GUARDAMOS BC EN EL STACK
3210 ;
3220 LD HL,22528 ; HL = PRIMERA DIRECCION DE LOS ATTRIB.
3230 LD B,A ; B = NUMERO DE LINEAS A MOVER
3240 ;
3250 SU2-AB ADD HL,DE ; HL = HL + DE
3260 DJNZ SU2-AB ; B = B - 1:IF B<>0 THEN SU2-AB
3270 ; HL = DIRECCION DE LA SEGUNDA LINEA
3280 ;
3290 PUSH HL ; GUARDAMOS HL EN EL STACK
3300 ADD HL,DE ; HL = HL + DE
3310 EX DE,HL ; SWAP HL,DE
3320 ; DE = DIRECCION DE LA PRIMERA LINEA
3330 POP HL ; HL = DIRECCION DE LA SEGUNDA LINEA
3340 POP BC ; BC = NUMERO DE BYTES A MOVER
3350 LDDR ; LD (DE), (HL):DE=DE-1:HL=HL-1:BC=BC-1
3360 ;
3370 LD A,(ATTR) ; A = ATRIBUTO A INTRODUCIR
3380 LD B,32 ; B = NUMERO DE CHR POR LINEA
3390 LD DE,22528 ; DE = DIRECCION DONDE PONER ATRIBUTOS
3400 ;
3410 CHR-AB LD (DE),A ; PONEMOS EL ATRIBUTO
3420 INC DE ; DE = DE + 1
3430 DJNZ CHR-AB ; B = B - 1:IF B<>0 THEN CHR-AB
3440 ;
3450 RET ; VUELTA AL BASIC

```

Inicialmente el programa ha sido localizado en la dirección 55000, pero como es reubicable, el usuario puede colocarlo donde mejor sirva a sus propósitos.

Con esta rutina podremos hacer **SCROLL**:

1. A la izquierda.
2. A la derecha.
3. Hacia arriba.
4. Hacia abajo.

Para decirle al programa en qué dirección queremos mover los atributos, sólo tenemos que poner el tipo de movimiento en la posición de memoria 23300. Si en ésta colocamos un número que no sea ni 1, ni 2, ni 3, ni 4, el programa volverá al BASIC sin hacer nada.

En la posición 23301 debemos poner el número de líneas que queremos que el pro-

grama «SCROLLee». Este número nunca puede ser mayor de 24.

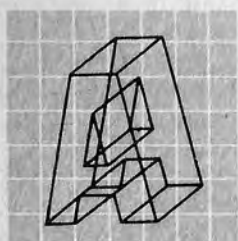
Como cada vez que se realiza un **SCROLL**, una línea o columna de la pantalla pierde los atributos que tenía, tenemos que poner en la posición 23302 el atributo que queremos que tenga la línea o columna que se va quedando vacía. Este número ha de estar comprendido entre 0 y 255. Los tres primeros BITS nos dirán el color de la tinta. Los tres siguientes el color del papel. El BIT (el número 6) dirá si queremos brillo (1) o no (0). Por fin, el último BIT nos dice si queremos **FLASH** (1) o no (0).

Una vez que se han **POKEado** los valores necesarios en todas estas direcciones, sólo hay que hacer:

RANDOMIZE USR 55000

para que la rutina funcione.

Conexiones en paralelo



DEMÁS de las conexiones de tipo serie presentadas en los últimos tomos, es posible conectar periféricos en modo paralelo utilizando la tarjeta de ampliación de puertos descrita en los primeros tomos. Existen, sin embargo, varios tipos de conexión utilizados en periféricos cuando la distancia de conexión es corta y que es interesante considerar.

Conexión Centronics

LADO CONTROL MACIZO

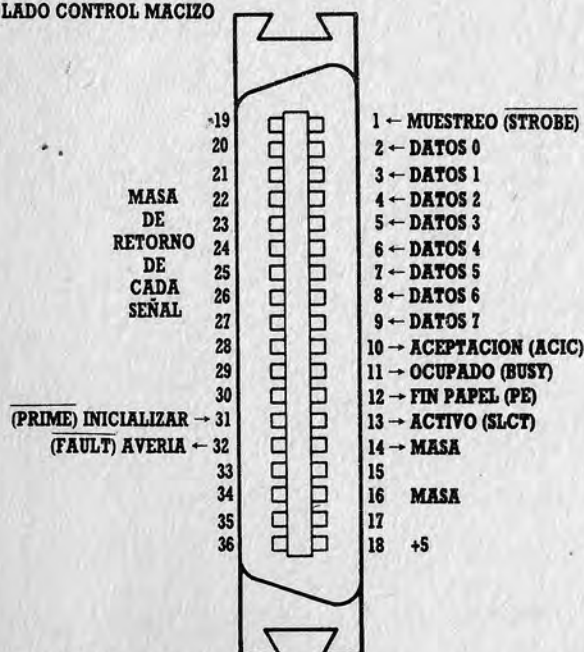


Fig. 1. Conector tipo Centronics.

Está diseñada para ser empleada principalmente en conexión de impresoras, pero es posible la adaptación de cualquier otro dispositivo utilizando la circuitería y protocolo ideados para transferencia en paralelo propios de la impresora. Utiliza 8 hilos de datos, y tres básicos de control: ocupado, aceptación y muestreo. En la figura se muestra la secuencia de señales necesaria para la transferencia. La comunicación es unidireccional, si bien en algunas realizaciones comerciales puede emplearse de modo bidireccional al poder desactivarse los circuitos de salida, de tipo Tristate. Los niveles de las señales en el conector son TTL, de donde se deduce la limitación de distancia a unos pocos metros, teniendo siempre en cuenta la adaptación de impedancias en el extremo receptor.

La interfaz paralelo de impresora del IBM-PC implementa el protocolo Centronics, aunque posee otros hilos para aplicaciones complementarias.

El conector es de 36 pines de tipo especial de borde. Usualmente el equipo receptor tiene el conector hembra, aunque mediante el cable de conexión se puede adaptar cualquier configuración. Por ejemplo, las impresoras del IBM-PC se conectan a través de un cable que tiene por un lado conexión a la tarjeta de adaptador paralelo, con conector de 25 pines de tipo Cannon hembra y por el lado de la impresora conector Centronics macho. La realidad es que son necesarios solamente 11 pines para el intercambio de señales, por lo que existen algunos equipos con conector reducido.

La velocidad de transferencia es del orden de 10.000 caracteres por segundo como máximo, por las necesidades del protocolo.

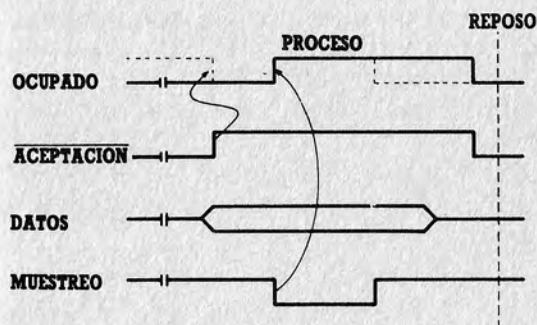
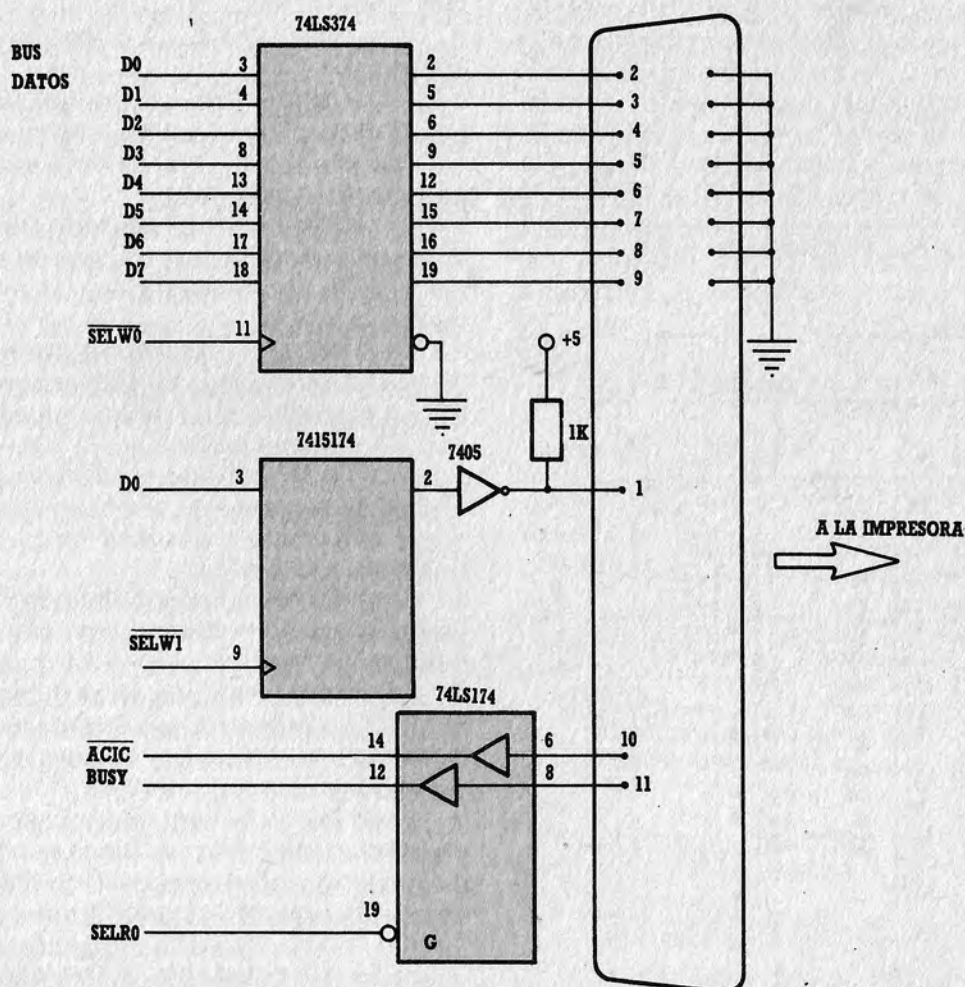


Fig. 2. Señales de la conexión Centronics.

Se puede montar una conexión de tipo Centronics a partir de una tarjeta de ampliación de puertos, o bien mediante la conexión paralelo de que disponen algunos equipos como el Commodore.



IEEE-488

Este tipo de conexión tiene su origen en el HP-IB, diseñado por Hewlett-Packard para comunicar diferentes instrumentos de medida con un ordenador. A veces se le denomina GP-IB, por General Purpose Interface Bus. Actualmente está muy difundido entre los equipos de instrumentos avanzados el disponer de forma estándar u opcional de la conexión IEEE-488. Requiere inteligencia en todos los equipos conectados a este bus. El protocolo de comunicación es muy complejo, por lo que daremos solamente unas indicaciones de sus posibilidades. Existe un documento formal del IEEE que hace la descripción completa. La gran difusión ha permitido el desarrollo de circuitos integrados que implementan el protocolo fabricados por Intel, Motorola y Texas. Por ejemplo, el circuito integrado Intel 8291A realiza todas las funciones de conexión al bus IEEE-488 como locutor u oyente (Talker/Listener), permitiendo el diseño simplificado de la conexión de un instrumento basado en microprocesador al bus. Las funciones de controlador pueden realizarse mediante el circuito Intel 8292, mientras que las de nivel físico pueden satisfacerse con el circuito Intel 8293. Con este conjunto de circuitos puede montarse la conexión completa para acceso directo a memoria de los datos transmitidos y recibidos a través del bus.

Este tipo de conexión ha sido también aceptado como norma de la IEC con el número 625-1, permitiendo un tipo diferente de conector, por lo que existen realmente dos normas en cuanto al conector externo se refiere.

Los datos son transmitidos por el bus en código ASCII de siete bits, pudiendo emplearse el octavo como de paridad.

La conexión entre los periféricos es en forma de bus, pudiendo tener una distancia entre extremos de 20 metros. El medio físico es un cable plano de 24 hilos con conectores especialmente diseñados para tener un extremo macho y otro hembra. A la salida del controlador los diferentes equipos pueden conectarse en cascada del mismo conector. Las señales deben cumplir niveles compatibles con TTL: 0 a 0,8 es nivel bajo o 0 y >2,5 voltios es nivel alto o 1. Todas las señales de control son activas en nivel bajo.

Las señales poseen la siguiente denominación:

— DIO1-DIO8 - Bus de datos. Bidireccional.

— DAV - Dato disponible. Es activada por el dispositivo que está controlando el bus en ese momento para indicar que los datos permanecen estables.

— NRFD - No preparado para datos. La activa el receptor para indicar que no está en disposición de recibir más datos por el momento.

— NDAC - Dato no aceptado. Es activada por el receptor para indicar que los datos deben mantenerse en el bus, pues todavía no han sido almacenados.

— ATN - Atención. Señal activada por el dispositivo que hace de controlador para indicar a los demás que hay un mensaje en el bus para todos ellos.

— IFC - Limpiar la interfaz. Sirve para poner a todos los dispositivos en condiciones iniciales.

— SRQ - Petición de servicio. Es utilizada por los equipos no controladores para indicar la solicitud de servicio por parte del controlador.

— REN - Activar control remoto. Se emplea para desactivar las funciones locales que pudiera tener el dispositivo direccionado, como, por ejemplo, el panel frontal de mandos.

— EOI - Fin o identificación. Si el que lo utiliza es el controlador, indica a los dispositivos que han pedido servicio que se identifiquen. Si el que lo utiliza es un dispositivo lo-

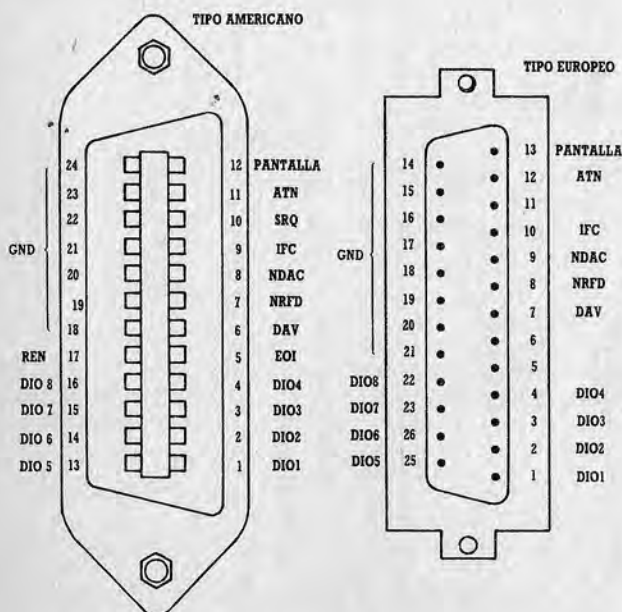


Fig. 4. Conectores IEEE-488.

cutor, significa que termina la transmisión de un bloque de datos.

Las transferencias se realizan con protocolo asíncrono, pero por los tiempos mínimos asignados a las señales resulta una velocidad máxima de transferencia del orden de 1 Megabyte por segundo. En todo momento debe haber un controlador del bus, que puede ser cualquiera de los equipos conectados. Cada dispositivo está identificado mediante una dirección, que puede ser de 6 o 10 bits, según se use el modo normal o el extendido. Existe doble juego de direcciones para dos tipos de función existente: locutor u oyente.

Los equipos pueden poseer diferentes posibilidades de operación, que deben ser definidas cuando se configura y que afectan a cómo se comportará en el seguimiento del protocolo. Podemos distinguir las siguientes posibilidades de operación:

- Controlador. Realiza las funciones de control del bus: autorización de peticiones de comunicación entre estaciones y envío de órdenes de activación y desactivación de dispositivos. Debe haber por lo menos un dispositivo que actúe de controlador. Puede efectuar cambios sobre el estado de los otros dispositivos conectados.

- Locutor. Tiene la posibilidad de transmitir información a otros dispositivos del bus, cuando le sea autorizado por el controlador. Para poder ser seleccionado dispone de una dirección única, que al aparecer en el bus activa los circuitos de recepción.

- Oyente. Tiene la posibilidad de recibir mensajes del bus, cuando sea autorizado por el controlador. Cada dispositivo que pueda actuar como oyente dispondrá de una dirección única que le permita distinguirlo de los demás.

Además, existen otras funciones dentro del protocolo:

- Control sobre envío (SH). Es una función adicional a la de locutor y que capacita para controlar la recepción por parte de los oyentes del mensaje transmitido. Es una función necesaria para los dispositivos que necesitan controlar la duración del envío desde el transmisor.

- Control sobre recepción (AH). Capacita para el control de la señal de aceptación, para permitir la sincronización con el locutor. Indica la posibilidad por parte del dispositivo del control de la señal NRFD y permitir el re-

traso de la transferencia de un dato por no haber podido aceptarlo o almacenarlo.

- Petición de servicio (SR). Capacita para solicitar el uso del bus de forma asíncrona, mediante la señal SRQ. La señal se mantendrá hasta que el servicio haya sido satisfecho.

- Sondeo en paralelo (PP). Permite a un dispositivo mostrar un bit de estado al controlador sin que se haya seleccionado como locutor. Permite el sondeo rápido de posibles solicitantes del bus, con una sola instrucción. Es posible disponerlo en 8 dispositivos simultáneamente.

- Inicialización específica (DC). Si se dispone de esta función puede activarse al dispositivo a sus condiciones iniciales de forma individual o por grupos.

- Disparo de dispositivos (DT). Permite la puesta en marcha de funciones específicas de un dispositivo, de forma individual o por grupos.

- Control remoto o local (RL). Mediante esta función puede controlarse la activación de funciones de forma local o solamente remota.

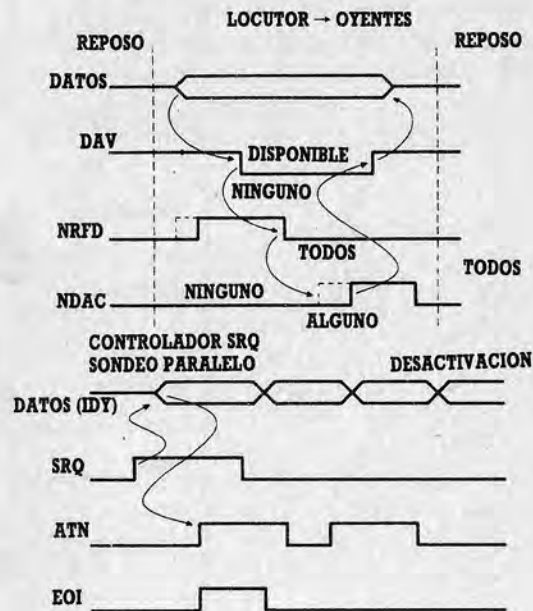


Fig. 5. IEEE-488 secuencia de intercambio de señales.

La secuencia normal de comunicación puede representarse con la figura con la descripción siguiente:

Para solicitud de utilización del bus:

- El dispositivo que quiere comunicar, activa la señal SRQ.

EL TALLER DEL HARDWARE

— El controlador busca el origen de la petición de dos formas posibles: en paralelo o en serie. En paralelo: el controlador activa la señal IDY y de los 8 posibles dispositivos el que solicitó pone a cero su bit correspondiente en el bus de datos. El controlador detecta el más prioritario, inhibe a los demás y activa a los oyentes y al locutor de la solicitud. Al desactivar la señal ATN el locutor comienza la transmisión. En serie: el controlador responde a la petición con la secuencia: UNL (inhibir a los oyentes), SPE (comienzo de encuesta serie), MTAi (dirección de locutor posible), STB (estado del locutor encuestado). Cuando al responder uno de los locutores posibles indica en su estado que solicitó servicio, se acaba la encuesta con SPD. El locutor recibirá control para enviar lo que desee.

— Se pasa a fase de envío de caracteres.

Para envío de un carácter:

— El dispositivo locutor observa que el bus no está en uso.

— Coloca la información en el bus.

— Activa la señal DAV (dato disponible).

— Los oyentes activan su señal RFD para indicar que no admiten más datos.

— Cuando han leído el octeto del bus, activan la señal DAC.

— Según el tipo de operación, el locutor observa DAC y al detectar que todos han aceptado, retira su señal DAV.

— Al detectarlo los oyentes reactivan su señal RFD, indicando que aceptan más datos. Solamente cuando todos los oyentes han aceptado, el locutor puede enviar el siguiente dato.

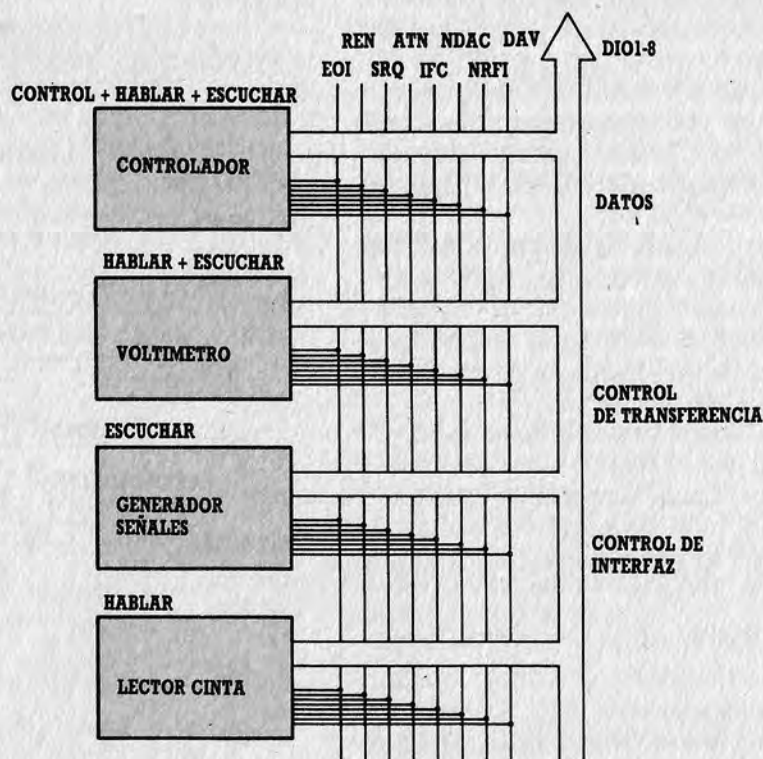


Fig. 6. Esquema general de conexiones.

La secuencia seguida para la transmisión de los mensajes cumple las condiciones del protocolo descrito. Además, las diferentes configuraciones de las señales existentes especifican diferentes tipos de mensajes. Los mensajes pueden estar constituidos por una o

más líneas. Se indican algunos ejemplos de mensajes para dar idea de la riqueza de posibilidades y también de la complejidad del soporte de este tipo de conexión. En cada mensaje individual las diferentes líneas de la interfaz deberán mantener los niveles corres-

pondientes a la señal activa. Los diferentes tipos de mensajes se dividen en varios grupos de comportamiento bastante homogéneo.

Grupo de mandatos universales: deben poder ser ejecutados por todos los dispositivos:

- Inicializar dispositivo (DCL).
- Bloqueo local (LL).
- Desactivar sondeo paralelo (PPU).
- Comienzo de sondeo serie (SPE).
- Fin de sondeo serie (SPD).
- Identificación (IDY).
- Inicializar la interconexión (IFC).
- Activar control remoto (REN).

Grupo de mandatos selectivos: solamente podrán ser ejecutados por aquellos dispositivos preparados para ello:

- Pasar a local (GTL).
- Activar sondeo paralelo (PPC).
- Inicializar los dispositivos seleccionados (SDC).

Control del direccionamiento:

— Envío de la propia dirección de oyente (MLA).

— Desactivar escucha (UNL). Mediante el envío de la configuración 11111, se desactivarán los oyentes seleccionados hasta ese momento.

— Envío de dirección de locutor (MTA).

— Desactivar locutor (UNT).

Grupo de mensajes secundarios:

- Dirección secundaria (MSA).
- Validación de sondeo en paralelo (PPE).
- Desactivación de sondeo en paralelo (PPD).

Grupo de mensajes de estado, para que los dispositivos indiquen su estado interno:

- Terminación (END).
- Estado (STB).
- Petición de servicio (SRQ).
- Respuesta de sondeo en paralelo (PPRi).

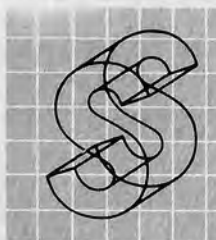
Grupo de mensajes de manipulación de datos:

- Datos aceptados (DAC).
- Dato válido (DAV).
- Preparado para recibir datos (RFD).

Para el IBM-PC existe una tarjeta para realizar conexiones IEEE-488 y el correspondiente programa de soporte de esta conexión desde cualquier lenguaje.

MATEMATICAS

Introducción a la topología



IN duda alguna, la topología forma uno de los pilares fundamentales en que se basan todas las matemáticas contemporáneas; actualmente es difícil imaginar el análisis o el cálculo infinitesimal sin la topología. Esta estudia

fundamentalmente las figuras y sus transformaciones en los espacios métricos; ¿y qué es un espacio métrico?; pues es un espacio en el que se ha definido una «distancia», hay que tener cuidado con este nombre, ya que aunque habitualmente la distancia que define un espacio métrico coincide con el concepto de distancia en la vida real, en otros muchos casos es así e incluso resulta imposible hacerse una idea intuitiva de lo que pueden significar. El concepto de distancia es muy importante en topología, ya que de él dependen los resultados de los teoremas posteriores. Una es una relación que tiene que cumplir los siguientes apartados:

1. Tiene que ser aplicación, y a cada par de elementos del espacio le corresponde un número real y sólo uno; por ejemplo, en la vida real la distancia habitual entre Madrid y Guadalajara es de 50 km.; este 50 es el número que asignamos a esa distancia, claro está que si vas por otro camino la distancia puede ser mayor, por ejemplo, si pasamos por Zamo-

ra antes de ir a Guadalajara, pero no importa, puesto que tomamos siempre la misma distancia independientemente del camino que se tome.

2. $d(x,y)=0$ si y sólo si $x=y$; que en castellano significa, a saber: que si la distancia de un punto a otro es cero, los puntos coinciden, o lo que es lo mismo, si dos puntos coinciden, su distancia es cero; claro, la distancia de Madrid a Madrid son cero kilómetros.

3. $d(x,y)=d(y,x)$; o sea, que la distancia entre Madrid y Barcelona es siempre la misma que desde Barcelona a Madrid.

4. $d(x,y) \leq d(x,z)+d(y,z)$; claro, la distancia entre dos puntos es igual o menor que la distancia entre el primer punto y otro intermedio y éste y el segundo punto, o sea, que si quiero tardar más en recorrer la distancia de Madrid a Barcelona lo mejor que puedo hacer es dar un rodeo y pasar por Valencia.

Con estos axiomas, el último no lo es, se definen las distancias y a partir de ellas todos los conceptos de la topología.

Pero no os desaniméis, la topología estudia también cosas más entretenidas como demostrar que una pelota de fútbol no se puede transformar en un donuts, pues el donuts tiene un agujero. También que si vas vestido con chaleco y chaqueta puedes quitarte el chaleco sin necesidad de quitarte la chaqueta, lo que demuestra que a efectos topológicos el chaleco está tan fuera de la chaqueta como los elefantes del zoo.

El programa siguiente os servirá de ayuda para comprender el importantísimo concepto de distancia y así introducimos poco a poco en el fascinante mundo de la topología.

```

10 REM *****
20 REM *
30 REM *      INTRODUCCION      *
40 REM *      A LA            *
50 REM *      TOPOLOGIA        *
60 REM *
70 REM *****
80 CLS
90 PRINT
100 PRINT "      N O C I O N      D E      D I S T A N C I A "
110 PRINT "      -----"
120 PRINT
130 PRINT
140 PRINT " DEFINICION:"
150 PRINT " -----"
160 PRINT "Se denomina distancia d, ";
170 PRINT "definida sobre E, a toda"
180 PRINT "aplicacion de ExE en R+, ";
190 PRINT "tal que a la pareja de"
200 PRINT "numeros x,y pertenecientes";
210 PRINT " a E, le asocia el numero "
220 PRINT "real no negativo d(x,y), ";
230 PRINT "que verifique las condi-"
240 PRINT "ciones siguientes:"
250 PRINT
260 PRINT "1)      d(x,y)=0 si y solo si x=y"
270 PRINT "      Axioma de separacion":PRINT
280 PRINT "2)      d(x,y)=d(y,x)      para todo x,y de E"
290 PRINT "      Axioma de la simetria":PRINT
300 PRINT "3)      d(x,y)<=d(x,z)+d(z,y) para todo x,y,z de E"
310 PRINT "      Desigualdad triangular"
320 PRINT
330 PRINT "Pulsa una tecla..."
340 x$=INKEY$
350 IF x$="" THEN GOTO 340
360 CLS
370 PRINT
380 PRINT
390 PRINT "Ahora veras una demostracion";
400 PRINT " grafica de los axiomas "
410 PRINT "anteriores. ";
420 PRINT "Para ello pulsa un numero"
430 PRINT "que se corresponda con ";
440 PRINT "el del axioma."
450 PRINT
460 PRINT "1.- Axioma de separacion"
470 PRINT
480 PRINT "2.- Axioma de simetria"
490 PRINT
500 PRINT "3.- Desigualdad triangular"
510 PRINT
520 PRINT "4.- Acabar"
530 PRINT
540 PRINT "Dime un numero del 1 al 4"
550 x$=INKEY$
560 IF x$="" THEN GOTO 550
570 IF VAL(x$)<1 OR VAL(x$)>4 THEN PRINT "Por favor, no te equivoques":GOTO 540
580 ON VAL(x$) GOSUB 640,870,1190
590 IF VAL(x$)<>4 THEN GOTO 360
600 '
610 REM FINAL DEL PROGRAMA
620 '
630 END
640 REM *****
650 REM * Axioma de separacion *
660 REM *****
670 '
680 CLS
690 LOCATE 20,10
700 PRINT "x"
710 LOCATE 20,11
720 PRINT ","
730 LOCATE 20,12
740 PRINT "y"
750 PRINT
760 PRINT
770 PRINT
780 PRINT "Si los puntos x e y coinciden ";
790 PRINT "entonces la distancia que les "
800 PRINT "separa es cero"
810 PRINT
820 PRINT

```

APRENDER CON EL ORDENADOR

```
830 PRINT "Pulsa una tecla..."
840 x$=INKEY$
850 IF x$="" THEN GOTO 840
860 RETURN
870 REM *****
880 REM * AXIOMA DE SIMETRIA *
890 REM *****
900 "
910 CLS
920 LOCATE 20,10
930 PRINT "x"
940 LOCATE 20,11
950 PRINT "."
960 LOCATE 20,12
970 PRINT "----->"
980 LOCATE 20,13
990 PRINT "  d(x,y)"
1000 LOCATE 20,9
1010 PRINT "<-----"
1020 LOCATE 20,8
1030 PRINT "  d(y,x)"
1040 LOCATE 32,10
1050 PRINT "y"
1060 LOCATE 32,11
1070 PRINT "."
1080 LOCATE 1,15
1090 PRINT "La distancia entre x e y ";
1100 PRINT "es la misma que la distan-"
1110 PRINT "cia de y a x, es decir ";
1120 PRINT "no importa el sentido en que"
1130 PRINT "se tome."
1140 LOCATE 1,20
1150 PRINT "Pulsa una tecla..."
1160 x$=INKEY$
1170 IF x$="" THEN GOTO 1160
1180 RETURN
1190 REM *****
1200 REM * desigualdad triangular *
1210 REM *****
1220 "
1230 CLS
1240 LOCATE 10,10
1250 PRINT "x          y"
1260 LOCATE 10,9
1270 PRINT "  d(x,y)"
1280 LOCATE 10,11
1290 PRINT "          ."
1300 LOCATE 15,17
1310 PRINT "          ."
1320 LOCATE 15,18
1330 PRINT "z"
1340 LOCATE 20,14
1350 PRINT "d(z,y)"
1360 LOCATE 4,14
1370 PRINT "d(x,z)"
1380 MOVE 75,225
1390 DRAW 40,-90
1400 DRAW 40,90
1410 DRAW -80,0
1420 LOCATE 1,20
1430 PRINT "En la figura se aprecia";
1440 PRINT " perfectamente que la"
1450 PRINT " distancia entre el punto ";
1460 PRINT "x y el y es menor tomada ";
1470 PRINT " directamente que pasando ";
1480 PRINT " por el punto z."
1490 x$=INKEY$
1500 IF x$="" THEN GOTO 1490
1510 RETURN
```

Este programa muestra los axiomas fundamentales del concepto de distancia; el mismo explica cómo funciona, así que obviaremos la explicación dejando al sagaz lector descubrirlo.

El programa funciona sin cambios en ordenadores AMSTRAD, en SPECTRUM cam-

biar la instrucción LOCATE por PRINT AT, y quitar todos los puntos y comas del final de las sentencias PRINT. En compatibles IBM, cambiar las variables de LOCATE (x,y) por LOCATE (y,x).

En COMMODORE cambiar las líneas que ponga:

```

340 x$=inkey$
350 if x$=<> then goto 350

```

por la siguiente:

```

340 get x$

```

También cambiar CLS por PRINT
CHR\$(147)

NATURALEZA Y TECNOLOGIA

■ Juego de la vida

El juego de la vida, también conocido como juego de vida, consiste en el estudio del comportamiento de unos entes llamados células, y su interacción con las circundantes.

Sus reglas son sencillas, en un espacio llamado universo, donde las células viven, se considera una célula y su entorno formado por las ocho adyacentes; estas células pueden tener dos estados: vivas y muertas; el juego con-

siste en determinar el estado de cada una de ellas en función de las que las rodean, para lo cual se aplican solamente tres reglas:

1. Si una célula está rodeada por una célula o menos, muere por aislamiento; si está rodeada por cuatro o más, entonces muere por asfixia.
2. Si tiene dos o tres rodeándola, entonces la célula conserva el estado en que estaba, bien muerta o viva.
3. Si está rodeada por tres células, entonces si la célula estaba muerta, nace una nueva.

Con estas tres reglas tan sencillas se calculan las generaciones que consisten en un «golpe de tiempo» en el que se calculan los estados celulares para todas y cada una de las células existentes, sin que su nuevo estado afecte el estado anterior.

De la configuración inicial de las células dependerá su desarrollo posterior; de esta forma encontramos configuraciones que son estables, que varían entre dos estados deslizadores que cada cierto tiempo se mueven por el universo, cañones que producen «balas» que se mueven por el universo, etc.

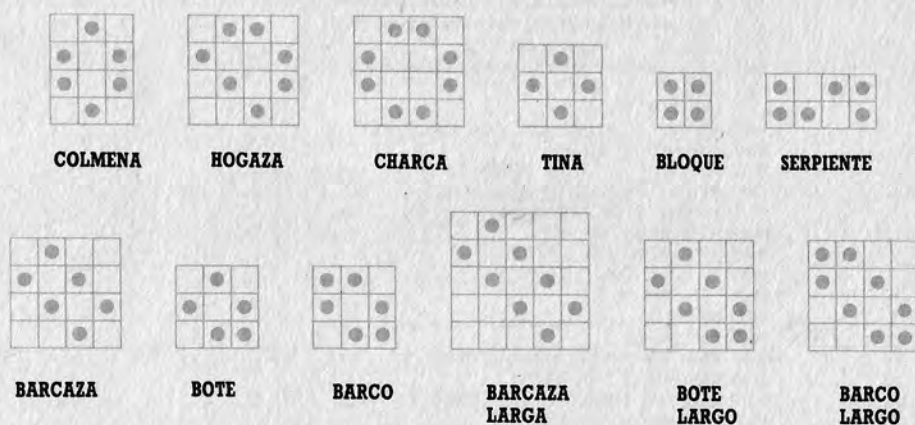


Fig. 1. Las formas estables más corrientes.

A pesar de su aparente simplicidad, el juego de la vida tiene innumerables aplicacio-

nes científicas y técnicas, en biología, botánica, ingeniería, y en otras muchas disciplinas.

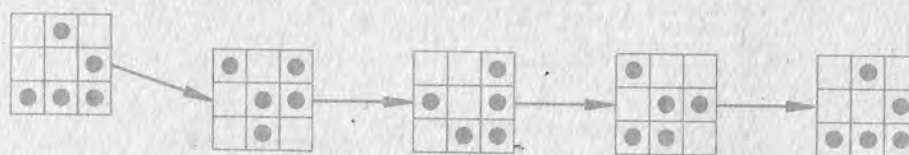


Fig. 2. El «deslizador».

APRENDER CON EL ORDENADOR

```

10 REM *****
20 REM *
30 REM *   JUEGO DE LA VIDA   *
40 REM *
50 REM *****
60 FILA=0: COLUMNA=0
70 NG=0: F=0: C=0
80 REM DECLARACION MATRICES
90 DIM CELU$(10,10), CELU1$(10,10)
100 *
110 REM PROGRAMA PRINCIPAL
120 *
130 CLS
140 LOCATE 10,10
150 PRINT "JUEGO DE LA VIDA"
160 GOSUB 430: LEER DATOS
170 GOSUB 800: IMPRIMIR CELULAS
180 LOCATE 1,23
190 PRINT "PULSA UNA TECLA"
200 X$=INKEY$: IF X$="" THEN GOTO 200
210 LOCATE 1,23
220 PRINT " "
230 *
240 REM BUCLE PRINCIPAL
250 *
260 GOSUB 1010: CALCULO
270 GOSUB 800: IMPRIMIR CELULAS
280 LOCATE 1,20
290 PRINT "PULSA UNA TECLA"
300 PRINT "PARA TERMINAR PULSA 'N'"
310 X$=INKEY$: IF X$="" THEN GOTO 310
320 LOCATE 1,20
330 PRINT " "
340 PRINT " "
350 IF X$<>"N" AND X$<>"n" THEN GOTO 240
360 *
370 REM FIN DEL PROGRAMA
380 *
390 CLS
400 END
410 * *****
420 * *****
430 REM *****
440 REM *   LEER DATOS   *
450 REM *****
460 CLS
470 LOCATE 10,1
480 PRINT "INTRODUCE LAS CELULAS"
490 LOCATE 1,20
500 PRINT " '* ' LAS PONE"
510 PRINT "ESPACIO LAS QUITA"
520 PRINT " 'F' PARA TERMINAR"
530 FOR I=1 TO 10
540   FOR J=1 TO 10
550     CELU$(I,J)=" "
560   NEXT J
570 NEXT I
580 F=1: C=1
590 REM COMIENZA LA LECTURA
600 LOCATE 35+C,7+F
610 LETRA$=INKEY$: IF LETRA$="" THEN 610
620 O=ASC(LETRA$)
630 IF O=50 THEN F=F+1: IF F>=10 THEN F=10
640 IF O=56 THEN F=F-1: IF F<=1 THEN F=1
650 IF O=52 THEN C=C-1: IF C<=1 THEN C=1
660 IF O=54 THEN C=C+1: IF C>=10 THEN C=10
670 IF O=42 OR O=13 THEN A$="X": GOSUB 1860
680 IF O=32 THEN A$=" ": GOSUB 1860
690 LOCATE 5,5
700 PRINT "FILA: "; F; "COLUMNA: "; C
710 IF O<>70 AND O<>102 THEN GOTO 590
720 LOCATE 10,1
730 PRINT " "
740 LOCATE 1,20
750 PRINT " "
760 PRINT " "
770 PRINT " "
780 REM FIN SUBROUTINA LEER DATOS
790 RETURN
800 REM *****
810 REM *   IMPRIMIR CELULAS   *
820 REM *****

```

```

830 CLS
840 G=G+1
850 LOCATE 5,1
860 PRINT "GENERACION NUMERO: ";G
870 LOCATE 37,8
880 PRINT "1234567890"
890 FOR I=1 TO 10
900 LOCATE 35,8+I
910 PRINT CHR$(47+I)
920   FOR J=1 TO 10
930     F=I+1
940     C=J+1
950     LOCATE 35+C,7+F
960     PRINT CELU$(I,J)
970   NEXT J
980 NEXT I
990 RETURN
1000 REM FIN SUBROUTINA IMPRIMIR CELULAS
1010 REM *****
1020 REM * CALCULO DE GENERACIONES *
1030 REM *****
1040 LOCATE 10,20
1050 PRINT "ESPERA UN MOMENTO, ESTOY PENSANDO."
1060 FOR I=1 TO 10
1070   FOR J=1 TO 10
1080     CELU1$(I,J)=" "
1090   NEXT J
1100 NEXT I
1110 HAY=0
1120 FOR C=2 TO 9
1130   HAY=0
1140   IF CELU$(1,C-1)<>" " THEN HAY=HAY+1
1150   IF CELU$(1,C+1)<>" " THEN HAY=HAY+1
1160   IF CELU$(2,C)<>" " THEN HAY=HAY+1
1170   IF HAY>4 OR HAY=1 THEN CELU1$(1,C)=" "
1180   IF HAY=2 OR HAY=3 THEN CELU1$(1,C)=CELU$(1,C)
1190   IF HAY=3 THEN CELU1$(1,C)="*"
1200   HAY=0
1210   IF CELU$(10,C-1)<>" " THEN HAY=HAY+1
1220   IF CELU$(10,C+1)<>" " THEN HAY=HAY+1
1230   IF CELU$(9,C)<>" " THEN HAY=HAY+1
1240   IF HAY>4 OR HAY<=1 THEN CELU1$(10,C)=" "
1250   IF HAY=2 OR HAY=3 THEN CELU1$(10,C)=CELU$(10,C)
1260   IF HAY=3 THEN CELU1$(10,C)="*"
1270 NEXT C
1280 HAY=0
1290 IF CELU$(1,2)<>" " THEN HAY=HAY+1
1300 IF CELU$(2,1)<>" " THEN HAY=HAY+1
1310 IF CELU$(2,2)<>" " THEN HAY=HAY+1
1320 IF HAY>4 OR HAY=1 THEN CELU1$(1,1)=" "
1330 IF HAY=2 OR HAY=3 THEN CELU1$(1,1)=CELU$(1,1)
1340 IF HAY=3 THEN CELU1$(1,1)="*"
1350 REM ESQUINA SUPERIOR DERECHA
1360 HAY=0
1370 IF CELU$(1,9)<>" " THEN HAY=HAY+1
1380 IF CELU$(2,9)<>" " THEN HAY=HAY+1
1390 IF CELU$(2,10)<>" " THEN HAY=HAY+1
1400 IF HAY>4 OR HAY<=1 THEN CELU1$(1,10)=" "
1410 IF HAY=2 OR HAY=3 THEN CELU1$(1,10)=CELU$(1,10)
1420 IF HAY=3 THEN CELU1$(1,10)="*"
1430 HAY=0
1440 IF CELU$(9,1)<>" " THEN HAY=HAY+1
1450 IF CELU$(9,2)<>" " THEN HAY=HAY+1
1460 IF CELU$(10,2)<>" " THEN HAY=HAY+1
1470 IF HAY>4 OR HAY<=1 THEN CELU1$(10,1)=" "
1480 IF HAY=2 OR HAY=3 THEN CELU1$(10,1)=CELU$(10,1)
1490 IF HAY=3 THEN CELU1$(10,1)="*"
1500 HAY=0
1510 REM ESQUINA INFERIOR DERECHA
1520 IF CELU$(9,9)<>" " THEN HAY=HAY+1
1530 IF CELU$(9,10)<>" " THEN HAY=HAY+1
1540 IF CELU$(10,10)<>" " THEN HAY=HAY+1
1550 IF HAY>4 OR HAY<=1 THEN CELU1$(10,10)=" "
1560 IF HAY=2 OR HAY=3 THEN CELU1$(10,10)=CELU$(10,10)
1570 IF HAY=3 THEN CELU1$(10,10)="*"
1580 REM PARTE INTERNA
1590 HAY=0
1600 FOR F=2 TO 9
1610   FOR C=2 TO 9
1620     HAY=0
1630     IF CELU$(F-1,C-1)<>" " THEN HAY=HAY+1

```

```

1640 IF CELU$(F-1,C)<>" " THEN HAY=HAY+1
1650 IF CELU$(F-1,C+1)<>" " THEN HAY=HAY+1
1660 IF CELU$(F,C-1)<>" " THEN HAY=HAY+1
1670 IF CELU$(F,C+1)<>" " THEN HAY=HAY+1
1680 IF CELU$(F+1,C-1)<>" " THEN HAY=HAY+1
1690 IF CELU$(F+1,C)<>" " THEN HAY=HAY+1
1700 IF CELU$(F+1,C+1)<>" " THEN HAY=HAY+1
1710 REM REGLAS DE EXISTENCIA
1720 IF HAY>4 OR HAY<1 THEN CELU1$(F,C)=" "
1730 IF HAY=2 OR HAY=3 THEN CELU1$(F,C)=CELU$(F,C)
1740 IF HAY=3 THEN CELU1$(F,C)="*"
1750 NEXT
1760 NEXT
1770 FOR F=1 TO 10
1780 FOR C=1 TO 10
1790 CELU$(F,C)=CELU1$(F,C)
1800 NEXT C
1810 NEXT F
1820 LOCATE 10,20
1830 PRINT "
1840 RETURN
1850 REM FINAL SUBROUTINA DE CALCULO
1860 REM PONER CELULA
1870 CELU$(F,C)=A$
1880 PRINT A$
1890 C=C+1
1900 IF C<=10 THEN GOTO 1940
1910 F=F+1
1920 C=1
1930 IF F>10 THEN F=1
1940 RETURN

```

El presente programa tiene una gran limitación en el número de células que puede manejar, ya que poniendo uno elevado la ejecución se hace excesivamente lenta.

Está estructurado en subrutinas para localizar más fácilmente los diversos pasos del cálculo de las generaciones, existiendo fundamentalmente tres, la de lectura de la configuración celular, la de impresión del universo, y, por último, la del cálculo de las generaciones.

El programa está adaptado a ordenadores AMSTRAD, en los que no habrá que realizar ningún cambio, en ordenadores compatibles con IBM, cambiar en los locates la coordenada X por la Y. En SPECTRUM recordar que las variables deben tener una sola letra y sustituir el LOCATE por PRINT AT. Si el ordenador es un COMMODORE, cambiar el CLS por PRINT CHR\$(147), y las líneas x\$=INKEY\$:IF X\$="" THEN GOTO por GET x\$.

SOCIEDAD

Filosofía. Lógica proposicional

Dentro del amplio campo de la filosofía, una de las menos conocida y estudiada es la lógica; y dentro de ésta a su vez es la lógica proposicional. Quizá el abandono de esta dis-

ciplina se deba a su entronque con las matemáticas y el álgebra de Boole, y a la separación que parece debe existir siempre entre ciencias y letras, que parece irreconciliable.

Vamos, pues, a repasar algunos conceptos de lógica proposicional y a intentar aclarar algo sobre el cálculo lógico y los operadores lógicos.

Para empezar recordemos lo que es una proposición; una proposición es un enunciado que puede ser verdadero o falso. La lógica se vale del simbolismo y así en vez de emplear enunciados de contenido los simboliza mediante letras; de modo y manera que en la lógica proposicional cada letra equivale a una proposición (se suelen usar las últimas letras, por ejemplo: p, q, r, etc.).

Las proposiciones pueden ser simples y compuestas; las simples están formadas por una sola proposición y las compuestas por dos o más proposiciones. También pueden dividirse en proposiciones de orden uno, orden dos, orden tres, etc., según el número de proposiciones simples que formen la proposición compuesta.

Una vez definidas las proposiciones, su simbolismo y sus tipos vamos a pasar a las operaciones lógicas. Las operaciones lógicas de orden uno son las más simples, se trata simplemente de enunciados que tengan uno de estos dos valores; a saber: falso o verdadero. Normalmente se simbolizan así:

- «1» = verdadero (V)
- «0» = falso (F)

Las operaciones lógicas de orden dos surgen al combinar dos proposiciones de orden uno. Las posibilidades que pueden darse son las siguientes:

p	q
0	0
0	1
1	0
1	1

Esta combinación se hace para que surjan proposiciones de orden dos que resultan de la unión (a través de ciertos operadores lógicos) de p y q; esta proposición resultante puede tener a su vez otros valores de verdad y falsedad. Vamos a estudiar algunas de las proposiciones principales de orden dos.

En primer lugar, estudiaremos la proposición disyuntiva inclusiva; es aquella proposición de orden dos que sólo es falsa si lo son las dos proposiciones que la integran. El símbolo es «v». Equivale a la suma lógica. Se puede leer como «o»; p o q o ambas.

p	q	p v q
0	0	0
0	1	1
1	0	1
1	1	1

Fig. 3.

La proposición conjuntiva es aquella proposición de orden dos que sólo es verdadera en caso de serlo las dos proposiciones que la forman. Su símbolo de enlace es «^»; y equivale al producto lógico. Se puede leer como «y»; p y q.

p	q	p ^ q
0	0	0
0	1	0
1	0	0
1	1	1

Fig. 4.

La proposición disyuntiva exclusiva la podemos definir como la expresión de orden dos que sólo es verdadera al ser verdadera una sola de las proposiciones que la forman (o

lo que es lo mismo, cuando hay un número impar de «1»). El símbolo es (v); se lee p o q, pero no ambas, equivale al OR exclusivo lógico.

p	q	p v q
0	0	0
0	1	1
1	0	1
1	1	0

Fig. 5.

Faltaría, por último, la proposición negativa, que no es propiamente de orden dos, pues sólo hay una proposición; pero no es simple al tener un signo conectivo, el de negación. Su símbolo es «~» colocado encima de la letra que hace referencia a la proposición; se lee «n o p». Niega la variable de entrada. Equivale a la complementación lógica.

p	$\bar{p}$
0	1
1	0

Fig. 6.

Dentro de la lógica proposicional destacan algunas leyes que vamos a repasar a continuación. Todas aquellas proposiciones tautológicas (es decir, que tiene todos sus valores de verdad a «1», o sea, verdadera en todos los casos) reciben el nombre de leyes de la lógica proposicional. Algunas de las leyes más importantes son las siguientes:

— **Leyes de idempotencia:**

$$p \vee p = p; p \wedge p = p$$

— **Leyes asociativas:**

$$(p \vee q) \vee r = p \vee (q \vee r); (p \wedge q) \wedge r = p \wedge (q \wedge r)$$

— **Leyes conmutativas:**

$$p \vee q = q \vee p; p \wedge q = q \wedge p$$

— **Leyes de absorción:**

$$p \vee (q \wedge p) = p; p \wedge (q \vee p) = p$$

— **Leyes distributivas:**

$$p \vee (q \wedge r) = (p \vee q) \wedge (p \vee r); p \wedge (q \vee r) = (p \wedge q) \vee (p \wedge r)$$

— **Leyes de De Morgan:**

$$(p \vee q) = \bar{p} \wedge \bar{q}; (p \wedge q) = \bar{p} \vee \bar{q}$$

Fig. 7.

Vistos estos conceptos fundamentales de la lógica proposicional, pasaremos a reali-

APRENDER CON EL ORDENADOR

zar un programa que nos recuerde los principales operadores lógicos. Hemos realizado el programa a modo de menú, con las opciones

de «O lógica», «Y lógica», «O exclusiva lógica» y «complementación lógica».

```
10 REM *****
20 REM * PROGRAMA FILOSOFIA *
30 REM * OPERADORES LOGICOS *
40 REM *****
50 CLS
60 PRINT:PRINT
70 PRINT TAB(15);" M E N U "
80 PRINT:PRINT
90 PRINT TAB(10);" 1 - O LOGICO "
100 PRINT TAB(10);" 2 - Y LOGICO "
110 PRINT TAB(10);" 3 - O EXCLUSIVA LOGICA "
120 PRINT TAB(10);" 4 - COMPLEMENTACION LOGICA "
130 PRINT TAB(10);" 5 - FIN "
140 PRINT:PRINT TAB(15);" PULSA OPCION DESEADA ";
150 INPUT X
160 ON X GOSUB 1000,2000,3000,4000,5000
170 PRINT:PRINT TAB(10);"PULSA UNA TECLA"
180 LET A$=INKEY$: IF A$="" THEN 180 ELSE 50
1000 REM *****
1010 REM * O LOGICO *
1020 REM *****
1030 CLS
1040 PRINT " P ";:PRINT TAB(15);" Q ";:PRINT TAB(27);" P v Q ":PRINT
1050 FOR I=0 TO 1
1060 FOR J=0 TO 1
1070 PRINT I,J,(I OR J)
1080 NEXT J
1090 NEXT I:PRINT:PRINT TAB(5);" O LOGICO"
1100 RETURN
2000 REM *****
2010 REM * Y LOGICO *
2020 REM *****
2030 CLS
2040 PRINT " P ";:PRINT TAB(15);" Q ";:PRINT TAB(28);" P ^ Q ":PRINT
2050 FOR I=0 TO 1
2060 FOR J=0 TO 1
2070 PRINT I,J,(I AND J)
2080 NEXT J
2090 NEXT I:PRINT:PRINT TAB(5);" Y LOGICO"
2100 RETURN
3000 REM *****
3010 REM * O EXCLUSIVA *
3020 REM * LOGICA *
3030 REM *****
3040 CLS
3050 PRINT " P ";:PRINT TAB(16);" Q ";:PRINT TAB(27);" P XOR Q ":PRINT
3060 FOR I=0 TO 1
3070 FOR J=0 TO 1
3080 PRINT I,J,(I XOR J)
3090 NEXT J
3100 NEXT I:PRINT:PRINT TAB(5);" O EXCLUSIVA"
3110 RETURN
4000 REM *****
4010 REM * COMPLEMENTACION *
4020 REM * LOGICA *
4030 REM *****
4040 CLS
4050 PRINT TAB(13);" "
4060 PRINT " P ";:PRINT TAB(13);" P "
4070 FOR J=0 TO 1
4080 PRINT J;: LET B=NOT J
4095 IF B = -1 THEN PRINT TAB(12);1
4097 IF B=-2 THEN PRINT TAB(12);0
4090 NEXT J
4100 PRINT:PRINT TAB(5);"COMPLEMENTACION"
4110 RETURN
5000 REM *****
5010 REM * OPCION FIN *
5020 REM *****
5030 CLS:PRINT:PRINT:PRINT
5040 INPUT "ESTAS SEGURO DE QUERER ACABAR (S/N) ";R$
5050 IF R$="S" OR R$="s" THEN 5060 ELSE 50
5060 PRINT:PRINT:PRINT " * F I N * " :END
```

El programa funciona en ordenadores IBM y compatibles, en AMSTRAD, MSX, y SPECTRUM.

En COMMODORE cambiar:

— CLS por PRINT CHR\$(147).

— INKEY\$ por GET X\$.

PARA LOS MAS JOVENES

Baloncesto

Vamos a tratar en esta sección de una disciplina que en los colegios no se suele to-

mar como fundamental, ésta es la educación física. Es una materia de las consideradas como «marías», es decir, de las que se tienen como fáciles y de segunda importancia. Así, pues, vamos a intentar hacer un elogio de ella y reconocer los méritos e importancia que tiene.

Ya desde antiguo, tanto griegos como romanos tuvieron una especial consideración a mantener el cuerpo ágil y fuerte, lo que hoy decimos «en forma»; podemos considerar que había un culto al cuerpo, como lo demuestran las estatuas clásicas que han llegado hasta

nuestros días. Basta con recordar algunas de las obras más famosas, como el *Discóbolo*, de Mirón; el *Doríforo*, de Policeto, que tiene el canon de un hombre fuerte, lo que hoy podíamos considerar casi un culturista; cualquiera de las obras de Fidias; el *Apolo*, de Belvedere, etcétera.

Así, pues, nuestras ansias de estar en forma no son nuevas, podemos aprovechar el dicho romano: «Nada es nuevo bajo el sol»; pues como ya hemos demostrado, en toda la Antigüedad clásica había un culto al cuerpo y una gran preocupación por la belleza.

Aprovechando otro refrán romano (y esta vez dicho en romano auténtico, o sea, en latín): «Mens sana in corpore sano»; que en traducción libre y parafraseando podemos decir que hace falta un cuerpo sano para tener un buen rendimiento del intelecto.

Visto esto, tenemos que reconsiderar la postura que se toma normalmente hacia la educación física en la enseñanza actual; y decir que ha de ser una de las asignaturas fundamentales.

Con este fin y viendo que en nuestros días se vuelve a poner de moda la preocupación por el cuerpo, vamos a realizar el primer programa deportivo de la historia de esta publicación.

Teniendo en cuenta el auge que está tomando el baloncesto en nuestros días, hemos decidido dedicar este programa a ese espectacular deporte, puesto que cada vez se practica más, y nuestro país mejora a grandes pasos tanto a nivel de clubes como de selección nacional.

El programa sirve para controlar los porcentajes que logramos durante un partido, así podemos mejorar el tiro, ver los rebotes capturados, cuántas asistencias hemos dado, qué tanto por ciento hemos anotado desde fuera de la línea de 6,25, cuántos tapones ponemos, etc.

Nombre	2P%	3P%	REB	ASIS	TAP	PTOS
EPI	100	66	4	2	2	36
ROMAY	83	100	9	0	11	13
MARGALL	100	85	2	5	0	36
MONTERO	77	100	2	5	0	38
ARCEGA	100	66	2	5	1	14
OK						

1 LIST 2 RUN 3 LOAD» 4SAVE» 5CONT 6, «LPT1 7TRON
8TROFF 9KEY OSCREEN

Fig. 8. Pantalla que sale en la versión 1.0.

Hemos realizado dos versiones del programa. La primera la podríamos llamar versión muestra; en ella hemos colocado en líneas DATA los nombres de cinco jugadores de la selección nacional de baloncesto para poner sus porcentajes y que sirva como muestra. Sólo hemos puesto cinco para que no se hiciera demasiado largo el programa, pero si se quiere seguir un partido de baloncesto entero con todos los jugadores, basta con dimensionar las matrices al principio con diez o doce posiciones y cambiar los datos almacenados en las líneas DATA por los nombres de los jugadores de nuestro equipo favorito.

```

4 REM *****
5 REM * PROGRAMA BASKET (VERSION 1.0) *
6 REM * SELECCION NACIONAL *
7 REM *****
8 REM
10 REM *****
20 REM * B A S K E T *
30 REM *****
40 REM
50 FOR I=1 TO 10
60 CLS
70 PRINT:PRINT:PRINT
80 PRINT TAB(15);"===== "
90 PRINT TAB(15);"XXXXXXXXXXXX "
100 PRINT TAB(15);"XXXXXXXXXXXX "
110 PRINT TAB(15);"XXXXXXXXXXXX "
120 PRINT TAB(15);"XXXXXXXXXXXX "
130 PRINT TAB(15);"XXXXX "
140 PRINT TAB(15);"XXXX "
150 PRINT:PRINT:PRINT
160 PRINT TAB(22);"C A N A S T A "
164 PRINT
165 PRINT TAB(25);"C A N A S T A "
170 FOR T=1 TO 250:NEXT T
180 NEXT I
190 CLS
200 REM *****
210 REM * INICIALIZAR VARIABLES *
220 REM * Y *
230 REM * DIMENSIONAR MATRICES *
240 REM *****
250 DIM TTD(5),TAD(5),TPD(5)
260 DIM TTT(5),TAT(5),TPT(5)

```

APRENDER CON EL ORDENADOR

```

270 DIM REB(5),TAPO(5),PTS(5),ASIS(5)
280 DIM JUG$(5),CAM$(7)
290 REM
300 REM *****
310 REM * INTRODUCCION DE LOS DATOS Y *
320 REM * CALCULO DE LOS PORCENTAJES *
330 REM *****
340 FOR I=1 TO 5
350   READ JUG$(I)
360   PRINT "TIROS DE 2 PTOS. DE ";JUG$(I);:INPUT TTD(I)
370   PRINT "TOTAL ACIERTOS DE 2 PTOS. DE ";JUG$(I);:INPUT TAD(I)
380   PRINT "TIROS DE 3 PTOS. DE ";JUG$(I);:INPUT TTT(I)
390   PRINT "TOTAL ACIERTOS DE 3 PTOS. DE ";JUG$(I);:INPUT TAT(I)
400   LET TPD(I)=INT((TAD(I)*100)/TTD(I))
410   LET TPT(I)=INT((TAT(I)*100)/TTT(I))
420   INPUT "REBOTES ";REB(I)
430   INPUT "ASISTENCIAS ";ASIS(I)
440   INPUT "TAPONES ";TAP(I)
450   LET PTS(I)=(TAD(I)*2)+(TAT(I)*3)
460 CLS
470 NEXT I
480 REM
490 REM *****
500 REM * PRESENTACION DE LOS RESULTADOS *
510 REM *****
520 REM
530 REM * LECTURA DE LOS DATOS *
540 FOR L=1 TO 7
550   READ CAM$(L)
560   PRINT TAB(6*L-1);CAM$(L);
570 NEXT L
580 PRINT
590 FOR S=1 TO 45
600   PRINT "=";
610 NEXT S
620 REM
630 FOR I=1 TO 5
640   PRINT JUG$(I);:PRINT TAB(10);TPD(I);:PRINT TAB(16);TPT(I);
650   PRINT TAB(23);REB(I);:PRINT TAB(29);ASIS(I);
660   PRINT TAB(35);TAP(I);:PRINT TAB(41);PTS(I)
670 PRINT
680 NEXT I
1000 REM *****
1010 REM * DATOS *
1020 REM *****
1030 DATA "EPI", "ROMAY", "MARGALL", "MONTERO", "ARCEGA"
1040 DATA "NOMBRE", "2P%", "3P%", "REB", "ASIS", "TAP", "PTOS"

```

La segunda versión es más personalizada, es decir, sirve para tu equipo de amigos o el equipo del colegio; en ella al principio dimensionas la matriz con el número de jugadores que participéis y luego vas introduciendo los nombres de los jugadores a través de un

INPUT; también los puedes almacenar en líneas DATA si quieres.

Bueno, que tengas suerte en el partido y a ver si llegas a ser un Magic Johnson, un Ralph Sampson, un Abdul Jabbar, etc.

```

10 REM *****
20 REM * PROGRAMA BASKET *
30 REM * VERSION 2.0 *
40 REM * TU EQUIPO *
50 REM *****
60 REM
70 REM *****
80 REM * S A S K E T *
90 REM *****
100 REM
110 FOR I=1 TO 10
120 CLS
130 PRINT:PRINT:PRINT
140 PRINT TAB(15);"XXXXXXXXXXXXX"
150 PRINT TAB(15);"XXXXXXXXXXXXX"
160 PRINT TAB(15);"XXXXXXXXXXXXX"
170 PRINT TAB(15);"XXXXXXXXXXXXX"
180 PRINT TAB(15);"XXXXXXXXXXXXX"
190 PRINT TAB(15);"XXXXXXXXXXXXX"
200 PRINT TAB(15);"XXXXXXXXXXXXX"
210 PRINT:PRINT:PRINT
220 PRINT TAB(22);"C A N A S T A"
230 PRINT
240 PRINT TAB(25);"C A N A S T A"
250 FOR T=1 TO 25:PRINT T
260 NEXT T
270 CLS
280 INPUT "NUMERO DE JUGADORES ";N
290 REM *****
300 REM * INICIALIZAR VARIABLES *
310 REM * Y *
320 REM * DIMENSIONAR MATRICES *
330 REM *****
340 DIM TTD(N),TAD(N),TPD(N)
350 DIM TTT(N),TAT(N),TPT(N)
360 DIM REB(N),TAPO(N),PTS(N),ASIS(N)
370 DIM JUG$(N),CAM$(7)

```

```

380 REM
390 REM *****
400 REM * INTRODUCCION DE LOS DATOS Y *
410 REM * CALCULO DE LOS PORCENTAJES *
420 REM *****
430 FOR I=1 TO N
440 INPUT "NOMBRE DEL JUGADOR ";JUG$(I)
450 PRINT "TIROS DE 2 PTOS. DE ";JUG$(I);:INPUT TTD(I)
460 PRINT "TOTAL ACIERTOS DE 2 PTOS. DE ";JUG$(I);:INPUT TAD(I)
470 PRINT "TIROS DE 3 PTOS. DE ";JUG$(I);:INPUT TTT(I)
480 PRINT "TOTAL ACIERTOS DE 3 PTOS. DE ";JUG$(I);:INPUT TAT(I)
490 LET TPD(I)=INT((TAD(I)*100)/TTD(I))
500 LET TPT(I)=INT((TAT(I)*100)/TTT(I))
510 INPUT "REBOTES ";REB(I)
520 INPUT "ASISTENCIAS ";ASIS(I)
530 INPUT "TAPONES ";TAP(I)
540 LET PTS(I)=(TAD(I)*2)+(TAT(I)*3)
550 CLS
560 NEXT I
570 REM
580 REM *****
590 REM * PRESENTACION DE LOS RESULTADOS *
600 REM *****
610 REM
620 REM * LECTURA DE LOS DATOS *
630 FOR L=1 TO 7
640 READ CAM$(L)
650 PRINT TAB(6*L-1);CAM$(L);
660 NEXT L
670 PRINT
680 FOR S=1 TO 45
690 PRINT "=";
700 NEXT S
710 PRINT:PRINT
720 REM
730 FOR I=1 TO N
740 PRINT JUG$(I);:PRINT TAB(10);TPD(I);:PRINT TAB(16);TPT(I);
750 PRINT TAB(23);REB(I);:PRINT TAB(29);ASIS(I);
760 PRINT TAB(35);TAP(I);:PRINT TAB(41);PTS(I)
770 PRINT
780 NEXT I
1000 REM *****
1010 REM * DATOS *
1020 REM *****
1030 DATA "NOMBRE", " 2P%", "3P%", "REB", "ASIS", "TAP", "PTOS"

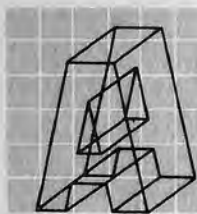
```

El programa funciona sin problemas en equipos IBM y compatibles, AMSTRAD, MSX, y SPECTURM.

En ordenadores COMMODORE poner en vez de CLS, PRINT CHR\$(147).

PEQUEÑA HISTORIA DE LA INFORMATICA

■ Los primeros ordenadores



partir del EDVAC, durante los años cuarenta, cincuenta y sesenta han ido apareciendo muchas máquinas, basadas en él.

Los primeros organismos que dispusieron de ordenadores fueron, como

cabe esperar, Universidades, principalmente americanas, aunque también británicas. Los grandes calculadores, complicados, fueron decayendo poco a poco hasta desaparecer. Sin embargo, para centrarnos en el embrollo de nuevas máquinas que fueron apareciendo después de la presentación del EDVAC, se pueden establecer varios períodos, marcados por acontecimientos relevantes, que han cambiado el rumbo de la informática.

Hasta 1951, todos los ordenadores que fueron creándose eran máquinas carísimas. Sólo grandes organismos, Estado, Universidades, etc., podían permitirse gastar millones de dólares en una máquina, que si bien realizaba tareas que hasta el momento no habían podido acometerse, en la mayoría de los casos su costo no las hacía rentables en absoluto. En estos años, poca gente creía en que las «hijas» de esas máquinas invadirían todas las tareas imaginables, llegando a los hogares domésticos, donde el niño de diez años las utilizaría para cálculos y juegos, el ama de casa, para llevar su pequeña contabilidad y realizar la declaración de la renta. Hasta el momento, ni la IBM se había interesado especialmente por ellas, pues Thomas Watson (su director) había quedado decepcionado tras su colaboración en el diseño del Mark I.

A partir de 1951, la mentalidad de los fabricantes cambia, y algunos comienzan a de-

cidirse a comercializar sus máquinas. Estas van cambiando progresivamente para adaptarse a las necesidades concretas de las industrias que las compran, pero siguen estando influenciadas por su nacimiento, al abrigo de las necesidades del Ejército, principalmente.

Volvamos, sin embargo, a 1945. El EDVAC acaba de aparecer dejando obsoleto a

¿Sabía usted que...

¿Sabía usted por qué se tardó tanto tiempo en fabricar el ordenador EDVAC? No, no se debió a problemas técnicos, que iban resolviéndose en los tiempos previstos. El retraso se debió a las enormes desavenencias que surgieron entre sus creadores, Eckert, Mauchly y Von Neumann. El problema principal, naturalmente, fue el económico. Eckert y Mauchly eran dos norteamericanos prácticos, para los que el trabajo realizado con gusto tiene un valor, pero que además es necesario, si se puede, obtener de él los mejores beneficios. En los primeros momentos, cuando todo lo que tenía el equipo eran unas ideas magníficas, no había desunión, sólo problemas por resolver. Al ir pasando el tiempo, consiguiéndose los objetivos previstos, Eckert y Mauchly consideraron las posibilidades del proyecto, desde el punto de vista comercial. Naturalmente, para obtener beneficios debía constituirse una sociedad, en la que participara el equipo, y que se ocupara de dar forma al problema comercial que surgía. Los restantes miembros del equipo, a la cabeza de los cuales se encontraba Von Neumann, tenían otros intereses muy distintos. Lo que deseaban era dedicarse a seguir investigando, bajo los auspicios de la Universidad. Mientras Eckert y Mauchly se interesaban por crear una nueva compañía, ocuparse de patentes, etcétera. Von Neumann estaba mucho más interesado en involucrar en sus trabajos a la Universidad, di-

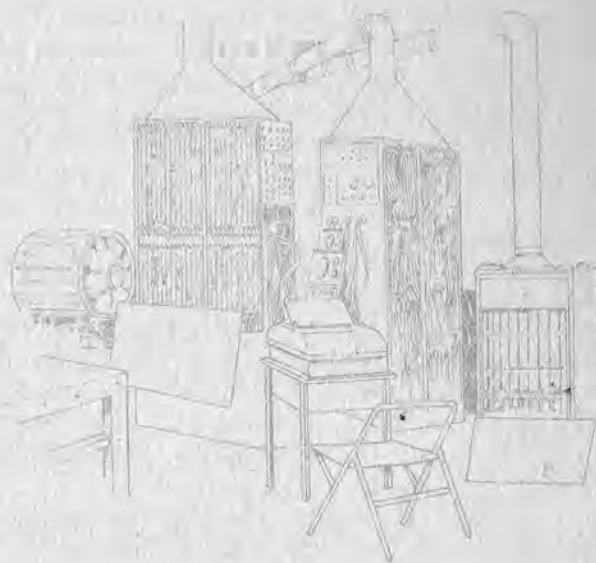
fundir los diseños, dar conferencias, etc. El trío, tanavenido en los comienzos de 1945, era ahora absolutamente incompatible.

Tras dos años de luchas intestinas, en las que entre los temas a discutir, uno de los más candentes era la paternidad del diseño, en abril de 1947, la justicia emitió su fallo. La compañía privada no podría crearse, ya que se consideraba que las patentes eran de dominio público. La verdad es que Von Neumann siempre había defendido como suyo propio un pequeño diseño previo de la máquina, en el que se establecían los principios básicos, aunque parece que la realidad es que, aunque estaba firmado por él, había sido creado entre los tres, Eckert Mauchly y él.

otro gran calculador, algo anterior a él, el ENIAC. Pero así como el EDVAC desbancó al ENIAC, hubo otros ordenadores basados en el EDVAC que aparecieron casi al mismo tiempo que éste, e incluso en Inglaterra varios años antes. En el recuadro anterior hablamos de las enormes divergencias existentes en el equipo que diseñó el EDVAC. Por un lado, el prestigioso profesor Von Neumann, deseando presentarlo al mundo científico, y por otro, los profesores Ecker y Mauchly, que preferían comercializarlo, y sacar de sus experiencias, ciencia e inteligencia un provecho material directo y sustancioso.) Estas querellas y desacuerdos retrasaron la fabricación de la máquina hasta 1951, y para entonces otros (mejor avenidos) habían aprovechado la lección. Entre las nuevas máquinas que aparecieron citaremos el IAS, BINAC, EDSAC, el Manchester MARK I y el proyecto ACE. Estos cinco ordenadores pueden considerarse los primeros. El ENIAC y el SSEC también pueden quizá incluirse dentro de la denominación de ordenador, aunque se trate de calculadores clásicos.

Von Neuman y Goldstine seguían manteniéndose unidos, afectivamente, y tenían contactos periódicos. En uno de estos contactos decidieron utilizar el ENIAC de un modo distinto, aprovechando algunas de sus tablas de función como memoria para almacenar las instrucciones. La idea era convertir el antiguo calculador en un ordenador. Sin embargo, un detalle fundamental distinguiría siempre el ENIAC de los ordenadores de arquitectura. Von Neumann: el ENIAC no disponía de unidad de control, pieza fundamental de cualquier ordenador.

Se hicieron los cambios precisos, e incluso se llegó a realizar una demostración, en



El ordeandor BINAC.

la que participó una especialista en programación, Adele, la esposa de Hermann Goldstine, pero la experiencia no pasó de eso, una pequeña práctica que ayudaría al equipo a ir comprendiendo mejor las limitaciones y ventajas de los distintos ordenadores. Veamos a continuación qué ocurrirá en Europa a mediados de siglo.

Entre los países que comenzaron a interesarse por la nueva ciencia en esos años destacó la Gran Bretaña. Este país acababa de terminar una guerra que le había dejado exhausta. Sin embargo, obtener dinero para investigación no era tan imposible como cabía esperar. Por un lado, durante la guerra se habían realizado experiencias con «calculadores» más o menos universales en el campo de la criptografía, ayudando a avanzar el fin de la contienda, y salvando numerosas vidas (aliadas, naturalmente). Por otro, el Ejército era popular y admirado, y había protegido con dedicación y eficacia a su nación. A pesar de sus dificultades económicas, el Gobierno decidió avanzar el programa nuclear, y, por tanto, necesitaba calculadores, programas de investigación e investigadores.

Las universidades comenzaron a salir de su letargo. Ya no estaban supeditadas a investigar sólo en determinados campos, sus profesores con prestigio podían investigar e incluso recibir ayuda del Gobierno, si sus trabajos eran considerados interesantes. A mediados de la década de los cuarenta, existía un plantel envidiable de matemáticos de gran

PEQUEÑA HISTORIA DE LA INFORMATICA

altura. Entre ellos McCulloch, Wilkes Womersley y Newman, padres de muchos de los proyectos de las nuevas máquinas universales. La mayoría de ellos había viajado a los Estados Unidos y conocía los trabajos de Von Neumann, Pitts y Turing, sin duda el de más prestigio en Inglaterra, a pesar de sus excéntricas.

Tres matemáticos de gran talla diseñarían tres prototipos británicos fundamentales, de los que sólo dos serían terminados. La calidad de estos trabajos sería, si no superior, al menos comparable a la alcanzada en los prototipos americanos. Veamos algo más sobre estos grandes investigadores, Womersley, Wilkes y Newman.

Uno de los centros de investigación británicos más importantes era el National Physical Laboratory, que había estado y continuaba en contacto con otros centros de investigación en los Estados Unidos de América. Entre sus miembros, el profesor Womersley estaba bastante interesado en los calculadores, y ya había trabajado en Cambridge y Manchester con analizadores diferenciales ingleses. También conocía la labor publicada de Turing (1936) y otros trabajos de matemáticos ingleses y americanos, implicados en proyectos de calculadoras.

Womersley fue invitado a los Estados Unidos para conocer los trabajos de EDVAC y MARK I.

Cuando volvió de los Estados Unidos estaba muy convencido del magnífico futuro de las nuevas máquinas, por lo que escribió a Turing muy admirado entonces por sus trabajos con este tipo de máquinas durante la guerra. Deseaba que le ayudara en la creación de un proyecto de ordenador inglés electrónico. Hasta ese momento, en Gran Bretaña sólo se habían creado las máquinas de Turing (durante la guerra) y una serie de grandes calculadores, los Colossus, que también utilizaban tubos electrónicos, aunque no eran totalmente máquinas universales. Estas máquinas (las Colossus) habían sido desarrolladas por Turing, junto con un gran matemático Max Newman, que era profesor de matemáticas en la Universidad de Manchester.

La nueva máquina que diseñó Turing, junto con el profesor Womersley, era realmente distinta a las demás. Se llamaría ACE (Automatic Computing Engine). Era rápida (un millón de pulsaciones por segundo), cinco veces superior al ENIAC. Su memoria era de 204.800 bits, también considerable. La máquina esta-



Hermann Goldstine.

ba diseñada basándose en la arquitectura del EDVAC, pero era innovadora y original. Su originalidad consistía principalmente en que además de disponer de algoritmos registrados en su interior, éstos se podían modificar durante la operación de la máquina por las propias instrucciones. La máquina resultó ser muy cara, y compleja. Además, y pese a una gran publicidad, hubo indecisiones entre sus financiadores, y como resultado, se terminó en 1950, sin la participación de su principal creador, Turing, que, aburrido de problemas no técnicos, se había marchado a Manchester con el profesor Max Newman.

Es una pena que los grandes creadores tengan tantos problemas para conseguir ver sus creaciones. Los ingleses habían hecho demasiada publicidad en torno a esa máquina «superior a los modelos americanos», según los principales diarios y radios británicos.

A mediados del año 1946, en la Universidad de Manchester, el profesor Max Newman había comenzado un proyecto de máquina de uso general, cuyo prototipo terminaría en 1948. El director del proyecto, F. Williams, había diseñado un nuevo tipo de memoria, basándose en tecnologías utilizadas en la generación de pantallas de radar e imágenes de televisión: el tubo de rayos catódicos. Los bits visibles en la pantalla se almacenaban en la me-

moria como puntos (impulsos eléctricos positivos o negativos). Además, dentro de la arquitectura de la máquina, existían otras seis memorias CRT que podían visualizarse en pantalla. Fue una revolución. Se trataba de una tecnología barata, rápida y que no necesitaba de grandes dimensiones. Las pantallas podían contener 2.048 bits o 1.024. Sin embargo, no era muy fiable, ya que algunos impulsos se transformaban o se perdían, con el correspondiente perjuicio.

El primer «manual de programación» fue también obra de Turing, y data de mediados del año 1948. Se trataba de un código para escribir los programas utilizando las teclas de un teletipo para representar las largas series de cifras binarias. De esas mismas fechas data el primer programa que fue ejecutado en un ordenador: 21 de junio de 1948. Williams introdujo el programa en la máquina y pulsó el botón correspondiente. «A partir de ese momento, comenzó una danza loca, una danza de muerte que no daba resultados utilizables..., pero en un momento dado se paró, y, brillando, reluciendo, en el lugar adecuado, aparecieron los resultados esperados» (son palabras de Williams, jefe del proyecto, que estaría naturalmente entusiasmado en ese momento histórico).

Otro centro de investigación de vanguardia era la Universidad de Cambridge, en

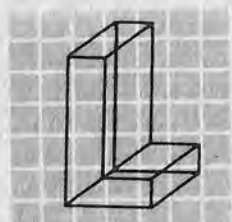
cuyo Laboratorio de Cálculo trabajaba el profesor Wilkes como director. El profesor Wilkes había asistido a un curso de verano en la Moore School, y conocía perfectamente los trabajos de Von Neumann. El tercer prototipo británico fue diseñado por el profesor Wilkes en su Universidad. Fue terminado algo más tarde que el Manchester MARK I, y se llamó EDSAC, «Electronic Delay Storage Automatic Calculator», calculador electrónico automático de memoria de retardo, porque disponía de una memoria de líneas de retardo de mercurio.

La máquina estaba basada en la arquitectura de Von Neumann del EDVAC. Disponía de unos 4.000 tubos de vacío y de la nueva memoria de líneas de retardo, de 32 líneas (se podían escribir cifras de treinta y dos dígitos con diecisiete decimales).

Este tipo de memorias fue inventado por Eckert y utilizado en algunos de los primeros ordenadores. Consistía fundamentalmente en que los impulsos eléctricos que representaban la información a almacenar circulaban por tubos y eran enviados de un punto a otro como si fueran un eco. Como este «eco» iba perdiendo intensidad con el tiempo, era activado regularmente, para mantenerlo en su estado inicial.

La máquina fue presentada en Manchester en 1949, y constituyó un gran éxito.

■ La informática aplicada a la producción industrial



A informática ha ido penetrando en la industria afectándola seriamente. De este modo, ha pasado a ser un problema político, patronal, e incluso sindical. El problema ha surgido debido a los cambios que ha provocado en la tecnología, pero principalmente por los producidos a nivel humano (tampoco debemos olvidarnos, sin embargo, de las desviaciones del capital, que pueden pasar a otros países si los productos de vanguardia informática se compran a empresas extranacionales y que acaban incidiendo en los problemas humanos).

Generalmente, los cambios tecnológicos se asocian a reducción de plantillas, y otros inconvenientes que aparecen casi siempre antes que las ventajas que producen dichos cambios. No revisaremos aquí esta visión simplista del problema, y nos limitaremos a ver los cambios que ocasiona la informática en la producción industrial.

En la empresa moderna los sistemas tradicionales de gestión de la mano de obra son absolutamente inadecuados para sacar partido de las nuevas tecnologías. No es posible organizar el trabajo tal y como se realizaba adecuación entre la tecnología avanzada que la gestión de stocks se llevaba a cabo manualmente, o mecanográficamente. En cualquier etcétera) y en muchos casos, una mayor adquisición de personal.

Es, pues, fundamental que exista una adecuación entre la tecnología avanzada que se va a utilizar y la organización del trabajo con esta tecnología. En muchos casos, si esta

adecuación no existe, las enormes inversiones que suelen suponer la adquisición de alta tecnología no tienen como resultado una mejora en la productividad, que, a fin de cuentas, suele ser la razón primordial de adoptar dichas tecnologías de vanguardia.

En tiempos anteriores, la industria utilizaba máquinas cuyos movimientos estaban controlados mediante relés electromecánicos. La gestión de stocks se llevaba a cabo manualmente, o mecanográficamente. En cualquier caso, los sistemas eran lentísimos, y además, utilizaban mucho espacio. A un mayor control, solía acompañar un mayor espacio de almacenamiento de información (armarios, ficheros, etcétera) y en muchos casos, una mayor adquisición de personal.

La increíble capacidad de cálculo de los ordenadores modernos, y sobre todo sus posibilidades de almacenar y utilizar en su momento cantidades ingentes de datos, los ha hecho perfectos como dispositivos de control de muchas máquinas herramienta, además de su labor algo más tradicional de manejo de stocks. Más aún, al ir bajando su precio y su volumen, se ha podido disponer de más potencia y capacidad de cómputo para un mismo precio, redundando en una mayor utilidad para muchos proyectos (algunos de los cuales, por su pequeño volumen, no podía pensarse en que fueran mecanizados hasta ahora).

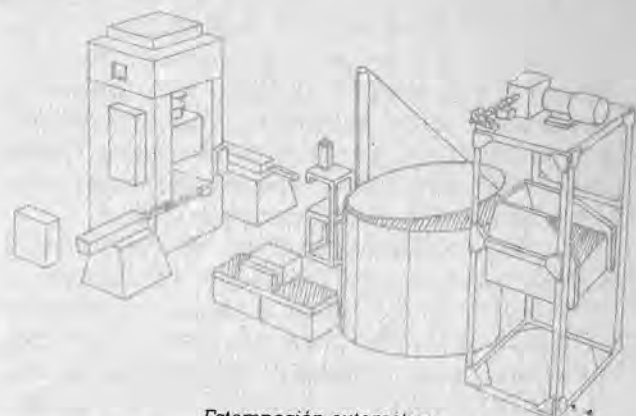
Otro punto a considerar es el proyecto de fabricación industrial de un determinado producto. La relación entre el proyecto y la fabricación posterior nunca ha sido tan estrecha. Hoy en día, antes de pasar a fabricar cualquier producto, se analizan cuidadosa y exhaustivamente sus condiciones de fabricación, montaje, almacenamiento, etc., para que éstos sean óptimos. (Es muy común fabricar elementos, considerando, por ejemplo, su forma de función de las dificultades de almacenamiento).

Tampoco debemos olvidarnos de la comunicación que debe existir entre los trabajadores o los equipos de fabricación y otros elementos del proceso de fabricación. Por lo general, las relaciones entre unos y otros componentes de la fábrica no eran anteriormente tan fluidas como cabía esperar. Aunque existían informes, éstos terminaban por limitarse a indicar condiciones o procesos extraordinarios, existiendo una tendencia generalizada a ir omitiéndolos, o al menos reservándolos para casos especiales.

Las condiciones tradicionales han tenido que irse abandonando paulatinamente si el industrial deseaba al menos sobrevivir. Las técnicas tradicionales empleaban mucha mano de obra, que no era tan cara como hoy en día.

Actualmente, a partir del flujo de información creado por la informática, que ha modificado las condiciones de producción industrial tan drásticamente, las piezas se diseñan partiendo de un software estándar, y existe una relación óptima entre el diseño-fabricación-control. Las operaciones suelen concentrarse en una máquina única, vigilada por personal muy especializado y flexible. La automatización va aumentando progresivamente, pero suele ser autorregulable, es decir, en caso de que alguno de los órganos de la máquina falle, el proceso procura continuar, bajo las nuevas condiciones especiales que se han producido. Japoneses y americanos han diseñado prototipos sin ningún tipo de vigilancia. Los resultados no han sido todo lo amplios que se deseaba. Normalmente, en trabajos complejos, los talleres sin vigilancia no han podido completar la totalidad de las tareas, quedando algunas reservadas para personal especializado. Además, el número de horas de trabajo sin supervisión que se consideró óptimo para la mayoría de los prototipos fue de ocho horas, mucho menos de lo que cabía esperar.

En una fábrica de automóviles, la sección que se ocupaba de la manufactura de cárteres de cajas de cambio debía fabricar dos receptáculos que unidos constituían la caja. Estos dos elementos debían fresarse, taladrarse y prepararse adecuadamente. La preparación de estas dos piezas comporta una decena de operaciones, cuyas tolerancias son además mínimas. Como hemos indicado antes, hace unos años estas operaciones estaban controladas por sistemas provistos de relés electromecánicos. Los sistemas modernos utilizan



Estampación automática

autómatas programables (es decir, máquinas gobernadas por un ordenador) que controlan los desplazamientos de los distintos brazos y equipo de la máquina con una precisión de centésimas de milímetro. Muchos de estos equipos van provistos de un sistema de mantenimiento que utiliza un equipo de vídeo, unido al autómatas. Cuando el equipo se para, o tienen algún problema, el sistema de mantenimiento indica el punto donde se produjo el fallo, su causa, deteniendo, si es necesario, la máquina, o las máquinas relacionadas con ella, que no podrían trabajar adecuadamente si ésta se detiene.

El sistema también controla otras operaciones, como el recambio de ciertas piezas, afilado, etc., cada cierto tiempo (puede ser cada 500, 700, 1.000 piezas que pasen por la máquina). Hasta el momento, es el personal especializado el que realiza las necesarias supervisiones de ciertos cambios de herramienta, e incluso otras pequeñas tareas que se realizan manualmente. Pero el trabajo de estas personas no consiste simplemente en pasearse por la unidad, comprobando que no existen problemas. Normalmente, en una planta de fabricación existen muchas máquinas, que requieren la atención y el trabajo de las personas que están a su cuidado, vigilando el material, y en otras pequeñas actividades complementarias, pero no por ello poco importantes.

Los sistemas actuales de fabricación han superado a los existentes hasta hace pocos años en unas cuotas increíbles del 50 al 70 por 100. (Esto significa que en una unidad de tiempo, los sistemas actuales fabrican de un 50 a un 70 por 100 más de piezas). Esta mejora tan espectacular está basada principalmente en la informatización de la industria. Las velocidades, precisión, autorregulación, han aumentado sustancialmente debido a las

condiciones de servicio a la máquina, y a las uniones entre los distintos procesos, calculadas para que sean las óptimas alcanzables, y que de este modo aumente considerablemente la producción. Se sabe mucho antes de su producción los costes exactos de cada producto según sea el proceso que sufrirá.

Las modificaciones de las condiciones tradicionales han hecho frágiles a los sistemas de producción. En muchos casos, las propias máquinas, o las herramientas que utilizan, resisten mal las velocidades de corte, o las cadencias propias de cada producción. Por otro lado, existe una enorme coordinación entre las operaciones de una máquina y entre una máquina y las restantes que integran una cadena de producción, lo que obliga a precisiones muy altas. El período de montaje en serie (cuya duración se suele multiplicar por dos, con respecto a los valores previstos) puede verse afectado negativamente por las múltiples modificaciones debidas fundamentalmente a errores de concepción. Es frecuente que estas modificaciones sean achacables a los problemas surgidos al informatizar la producción, mejorando velocidades, y, en general, obteniendo unas características muy superiores a las que eran frecuentes anteriormente. En muchas ocasiones no se consiguen alcanzar los ritmos de fabricación previstos porque las herramientas de las máquinas se desequilibran debido a determinados fallos dentro del proceso general. Este puede considerarse como una «construcción» o mecano, en el que unas piezas se apoyan en otras. Si una falla, o está colocada de forma imperfecta, las demás acusarán dicho fallo, funcionando con un rendimiento más bajo, o simplemente parándose y deteniendo el proceso (lo que ocasionará en muchos casos la detención de otras máquinas-proceso en cascada).

Este tipo de dificultades no significan en absoluto que el fabricante deba conformarse y remediarlas como pueda. En la práctica, son superables, y sólo indican que la informatización en la producción exige calidades y precisiones superiores a las aceptadas anteriormente. De todo lo expuesto, podemos deducir que las concepciones y diseños tradicionales no están generalmente a la altura de las necesidades de los nuevos sistemas de producción. Los ajustes técnicos y precisiones permitidas hace unos años no son aceptables con estas nuevas tecnologías, tan complejas y automatizadas. El maestro de taller que ve una pieza defectuosa no puede pararse a «arre-

glarla», o dársela al aprendiz (en este caso inexistente). Sus máquinas deben ser absolutamente fiables, ya que si existen errores, éstos no podrán subsanarse con un poco más de trabajo y buena voluntad.

Además de unas herramientas fiables, de calidades muy superiores a las utilizadas anteriormente, la cadena de producción debe disponer de un equipo formado durante el período de montaje que conozca perfectamente el funcionamiento del autómatas. También debe contar con otro equipo de mantenimiento que sea capaz de afrontar y solucionar con el mínimo tiempo posible las averías.

El fin principal de informatizar la producción es alcanzar cotas de productividad superiores. (La producción debe aumentar por unidad de capital —este incluye maquinaria, materia prima y salarios—.) Naturalmente, esta productividad está muy relacionada con las detenciones (de todo tipo) de la cadena (entre las que se cuentan cambios de herramientas, afilados, etc.). Por tanto, uno de los objetivos fundamentales es el máximo rendimiento de las máquinas (que no disponen de stocks tampón entre unas y otras). Los operadores directos, por tanto, tienen una tarea fundamental, y deben estar muy capacitados para que las detenciones sean mínimas. Como estamos viendo, alcanzar cotas de fabricación tan altas exige pequeños índices de error, altas calidades y altas cualificaciones del personal.

Con el fin de poder alcanzar las cotas previstas es frecuente disponer de varios equipos de personal especializado para una misma tarea. En muchas ocasiones, si una máquina concreta de la cadena deja de funcionar, sólo es posible tomar la determinación de alimentar las restantes de la cadena con un «stock ficticio», servido por los propios operadores. El resultado no será una detención absoluta, pero generará multitud de pequeños problemas, que posiblemente agraven la situación original.

Además de los problemas anteriores, deben considerarse otros aspectos del trabajo de los operadores. Este, en general, es repetitivo, y puede dar lugar a distracciones fatales (frecuentemente deben cambiar una determinada cuchilla, o colocar un elemento en un punto determinado, siempre el mismo durante toda la jornada). En general, las condiciones de trabajo se van deteriorando con el tiempo, al ir acostumbándose el trabajador al ritmo (cadencia) de trabajo de las máquinas. Sin embargo, los operarios, por otro lado, de-

ben reaccionar rápidamente ante las condiciones excepcionales, disponiendo de una capacidad de discernimiento y toma de decisiones adecuada a las instalaciones que está controlando.

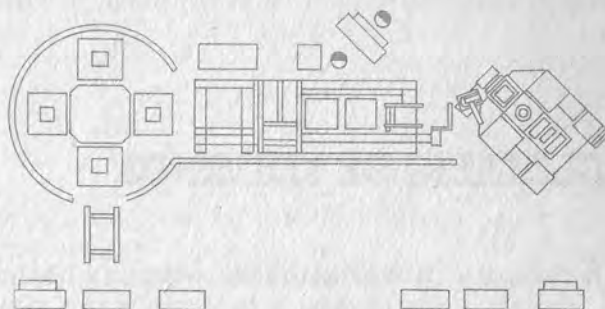
Los jóvenes son los que más se adecúan a este tipo de trabajos. Por un lado, no están acostumbrados a trabajar con sistemas tradicionales, a los que han tenido que adaptarse, para a continuación olvidarse de ellos, y por otro, suelen tener cualificaciones más altas que sus progenitores, estando mucho más acostumbrados que ellos a las innovaciones que van surgiendo en todos los campos.

Veamos muy sucintamente cómo suele repartirse el trabajo en estas líneas de producción:

- La responsabilidad de cada línea recae colectivamente sobre un grupo de operadores, bajo la dirección del maestro de taller. Cambian herramientas e intervienen en pequeñas averías, existiendo turnos rotativos para determinadas tareas.

- Es fundamental que los operadores sean polivalentes, es decir, sepan operar en la totalidad de las máquinas que componen la línea. También es importante que vayan rotando en el manejo de todas las máquinas, para que se mantengan al día en su funcionamiento.

- Las averías debe solucionarlas el equipo de mantenimiento, compuesto por personal especializado (en este equipo debe haber especialistas en fluidos, mecánicos, elec-



Línea de producción de moldes de todo tipo.

tricistas, especialistas en hardware, software, etcétera.

Resumiendo, el trabajo está organizado por especialistas, no por el personal que lo realiza. Este es polivalente y muy cualificado (no obedece exactamente a las normas vigentes hasta ahora, de división del trabajo y realización repetitiva de la misma tarea). Finalmente, la reparación de las averías es tarea de especialistas, mucho más preparados que los operadores para minimizar los tiempos de detención de las máquinas.

La informatización de la producción industrial exige un flujo creciente de información, y cualificaciones altas de los trabajadores. Esperemos que sirva para mejorar el nivel de vida y alcanzar cotas de confort más altas para los propios trabajadores. No debemos olvidar que no es lo mismo productividad que productivismo, y que la felicidad consiste en sentirse pleno, no en tener más.

GLOSARIO DE TERMINOS

Autómata programable. Máquina-herramienta automatizada y controlada por programa, es decir, la característica fundamental del autómata programable son las facilidades que ofrece para la modificación del programa (secuencia de tareas que ha de realizar la máquina). También suele designarse con el nombre de robot industrial.

Autorregulable, proceso. Las tareas que componen un determinado proceso de fabricación se dice que constituyen un sistema autorregulable cuando el conjunto está dotado de mecanismos mediante los cuales si se produce un fallo de algún elemento productivo o máquina, el sistema asigna automáticamente a esa tarea otros elementos productivos y/o herramientas, o elimina una de las fases de producción afectadas.

Fragilización de un sistema. Pérdida de la flexibilidad del conjunto productivo frente a la posible aparición de fallos, averías, etcétera. Si los sistemas productivos no están bien diseñados, la automatización industrial puede disminuir la flexibilidad del conjunto, haciendo más «frágil» el sistema.

Relé electromecánico. Dispositivo de control de tareas: la tarea básica asignada a un relé es la apertura o cierre automático de uno o varios circuitos eléctricos. En los procesos productivos, en general, los relés se suelen accionar mediante numerosos mecanismos: contacto físico, movimiento de una leva, cierre eléctrico de un circuito auxiliar, etcétera.

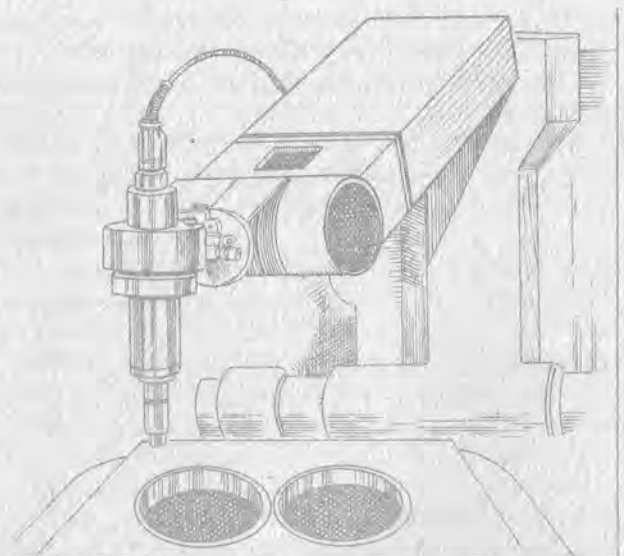
Stock. Almacenamiento. Conjunto de elementos almacenados bien sea bajo la forma de materia prima, como productos intermedios semielaborados, o como productos terminados.

Stock ficticio. En un sistema productivo en cadena, parte de los elementos de producción que se introducen en el sistema productivo desde el exterior de dicho proceso, manualmente, por parte de los operadores, cuando se produce una rotura del stock de

productos intermedios en un punto de la cadena. Suele producirse por una avería o desfase entre dos etapas de la cadena.

Stock tampón. Almacenamiento intermedio que se establece entre dos etapas de una cadena de producción cuando se prevé que pueden surgir discrepancias entre la capacidad de alimentación del stock de la cadena previa y las necesidades de consumo de la etapa subsiguiente.

Turnos rotativos. La tendencia actual en los sistemas de producción automatizados es sustituir la vieja norma de «especialización en el trabajo» por el nuevo concepto de «trabajadores polivalentes», capaces de asumir cualquiera de las tareas que hay que realizar en una etapa del sistema productivo. Este nuevo esquema de trabajo se completa con el establecimiento de turnos rotativos, que mantienen actualizados los conocimientos del trabajador, y evitan el cansancio que se solía producir cuando un operario realizaba durante períodos largos una misma tarea, de forma repetitiva.



Sistema de desbarbado de cajas de cambio en una factoría de automóviles.

VOCABULARIO DE INFORMATICA

No equal. Operador de relación que indica una desigualdad. Suele indicarse mediante los caracteres "< >" "> <" O "NE".

No lineal, programación. En investigación operativa, procedimiento para obtener el máximo o mínimo de una función, cuyas variables deben cumplir ciertas imposiciones. Dichas imposiciones o la propia función no son lineales (o ambas cosas). En contraposición a programación convexa, dinámica, entera lineal, matemática o cuadrática. También puede indicar una optimización no lineal.

Non polarized return to zero recording. Término inglés que indica un método de grabación en el que la condición de referencia es la ausencia de magnetización. De este modo, los unos se especifican mediante una condición de magnetización determinada, y los ceros por la ausencia de magnetización.

Non return to zero recording. Término inglés que indica un método de grabación en el cual el cambio en el estado de magnetización que representa el cero o el uno es el que especifica la condición de referencia.

NOR. Operador lógico que posee la siguiente propiedad: si P, Q y R son elementos simples, NOR de P, Q, R es verdad si todos los elementos son falsos, y falso si al menos uno so si P es verdadero. Su tabla de verdad es sentado con frecuencia mediante dos instrucciones: OR y NOT. Su tabla de verdad es la siguiente:

P	Q	P NOR Q
0	0	1
0	1	0
1	0	0
1	1	0

Normalizar. Multiplicar una variable (o una o más cantidades) por un coeficiente numérico para que otra cantidad asociada tome un valor dado. Sinónimo de escalar.

NOT. Operador lógico booleano que representa la negación. Si P es un elemento cualquiera, NOT será verdad si P es falso, y falso si P es verdadero. Su tabla de verdad es la siguiente:

P	NOT P
0	1
1	0

NOT-AND. Idéntico a NAND.

NOT-OR. Idéntico a NOR.

NOT-IF-THEN. Exclusión.

Notación. Sinónimo de representación. Indica los métodos utilizados para representar números o expresiones matemáticas.

VOCABULARIO DE INFORMATICA

Notación binaria. También llamado sistema binario. Notación en base dos. Es el sistema numérico más utilizado en Informática. Un bit puede ser 0 ó 1. La representación de los números utilizando dígitos binarios se llama notación binaria.

Notación en coma fija. Representación decimal con un formato tal que se reservan algunas posiciones para la parte entera del número, y otras (a partir de una posición establecida de antemano) para las cifras decimales.

Notación en coma flotante. Representación de un número en forma exponencial. En la representación en coma flotante se utilizan dos campos (aparte del campo de signo), uno donde se almacena un número decimal de la longitud necesaria para conseguir la precisión que se desea obtener (mantisa). El otro campo almacena el exponente al cual hay que elevar el número 10 para que la potencia resultante, multiplicada por la mantisa, sea el número (parte entera más parte decimal) que se desea representar.

Notación hexadecimal. Representación de los números en un sistema de base dieciséis. Los dieciséis dígitos hexadecimales suelen representarse mediante los números del 0 al 9 y las letras A a la F.

Notación polaca inversa. Forma especial de representar las expresiones algebraicas en la cual los operadores siguen a los valores numéricos. Por ejemplo, si se desea realizar la suma de $3 + 4$, se podría escribir $+ 3 \text{ ENTER } 4$. Con la notación polaca inversa, los operadores son los que se colocan al final, escribiéndose, por ejemplo, $"3 \text{ ENTER } 4 +"$. Este tipo de notación es utilizado en muchos modelos de calculadoras.

Núcleo magnético. Trozo de ferrita que se utilizaba en la construcción de memorias centrales magnéticas. Cada núcleo sólo almacenaba un bit. Este tipo de memorias se van sustituyendo actualmente por otros sistemas de memorias electrónicas. Sin embar-

go, la ventaja de las memorias magnéticas consistía en que al dejar de alimentar el ordenador no se perdía la información escrita en ellas (no eran volátiles).

Nulo, carácter. Carácter de control que sirve para rellenar espacios. En el código ASCII este carácter es distinto del cero numérico.

Numérico, carácter. Dígito.

Numérico, código. Código cuyos elementos son únicamente dígitos y caracteres especiales.

Numérico, control. Control automático de un proceso, realizado por un dispositivo que utiliza en todo o en parte los datos numéricos que le facilita un micro (introducidos generalmente como operaciones en el proceso).

Numérico, teclado. Teclado de las calculadoras. Se denomina así especialmente al pequeño teclado independiente (cuyas teclas están colocadas en el orden que suelen estar en las calculadoras de bolsillo), que existe en los ordenadores, separado del teclado general. Sólo contiene teclas numéricas.

Número real. En informática, un número compuesto por dos partes, una decimal y otra entera, separadas por una coma. Pueden ser representados en forma exponencial, mediante una mantisa y un exponente.

Números aleatorios. Sucesión de números que no guardan ningún tipo de relación unos con otros. Sinónimo de números obtenidos al azar. En casi todos los lenguajes informáticos existe una función para obtener números pseudoaleatorios (casi aleatorios).

Números complejos. Números constituidos por dos partes, una real y otra imaginaria. Surgieron para poder representar los resultados de determinadas operaciones, como, por ejemplo, las raíces cuadradas de los números negativos.

